Mapping Calendar Expressions into Periodical Granularities[*]

Claudio Bettini Sergio Mascetti
DICo - University of Milan,
via Comelico 39, I-20135 Milan, Italy

X. Sean Wang
Department of Computer Science
Univ. of Vermont, Burlington, VT, USA

Abstract

An effort has been devoted in the recent years to study and formalize the concept of time granularity and to design applications and services using the formalization. Among other proposals, a calendar algebra has been defined to facilitate the specification of new granularities and to perform conversions among them. This paper shows how granularities defined as algebraic calendar expressions can be represented as periodical sets of instants. More precisely, the paper shows how each algebraic operator changes the periodical structure of the granularities given as operands. These results have an immediate application enabling users to easily specify new granularities and using them in the only constraint solver supporting time granularities that is currently available.

1. Introduction

Temporal information in applications is often expressed differently for different purposes. For internal representation purpose, integers and integer intervals are a common choice. For example, in UNIX systems, time is expressed internally as the number of seconds elapsed since Jan 1, 1970. Other examples include temporal databases [EJS98], where integer intervals are often used internally as timestamps, and temporal contraint propagation [BWJ02], where sets of integer intervals are used.

On the other hand, when presented to users, temporal information is best expressed in terms of commonly used, organization-specific, or even user-specific calendars. A particular time such as “Wed Feb 11 21:56:32 EST 2004”, expressed in the common calendar, is much easier for a human user than “1,076,554,592 seconds since 00:00:00 1970-01-01 UTC” is, while the latter is easier for computers to deal with. The same holds for “every other Friday” versus a set of intervals of integers, each being a day, that gives all these Fridays.

In this paper, we consider the calendar algebra [NWJ02] as a symbolic representation tool at the user level. The algebra can be used to easily define commonly used granularities like day, week, month, and year, as well as specialized granularities like every other Friday, and first week of all semesters. We develop a method to convert calendar algebra expressions to internal integer interval representations (formally defined later). As a particular application of our results, users of the only system currently available for solving networks of temporal constraints with granularities (GSTP, [BWJ02]) can use calendar algebra expressions to define granularities appearing in the constraints.

The main contribution of the paper is that we show how to map an algebraic expression defining a time granularity into a periodic set representation of the same granularity. In particular, for each algebraic operator, the periodicity of the set corresponding to its application is derived from the periodicity of the argument expression. In the paper we also highlight a problem with the definition of the *altering tick* algebra operator in [NWJ02], and we propose a solution. Despite several formalisms have been proposed for symbolic representation of granularities and periodicity (among which [LMF86, Nie92, NWJ02, Ter03]), and some work has been done on comparing and enhancing the expressive power of some of them (e.g., [BD00]), no mapping was provided in these papers to identify how each operator changes the mathematical characterization of the periodicity of the argument expressions. The problem of finding these mappings is not trivial for some operators, and its solution enables interesting applications based on the mathematical characterization of periodic expressions.

In the next section we define granularities and discuss their internal symbolic representation. In Section 3 we outline our main task of conversion, identifying the periodicity for each algebraic operator. The derivation of the integer intervals corresponding to explicit granules in one period is shown in Section 4 and we conclude in Section 5.

[*] This work has been partially supported by Italian MIUR (FIRB ”Web-Minds” project N.RBNE01WEJT_005). Wang’s work has also been supported by US NSF under the career award 9875114.

2. Granularities and Their Symbolic Representation

Time granularities include very common ones like hours, days, weeks, months and years, as well as the evolution and specialization of these granularities for specific contexts or applications. Trading days, banking days, and academic semesters are just few examples of specialization of granularities that have become quite common when describing policies and constraints. A comprehensive formal study of time granularities and their relationships can be found in [BJW00]. In this paper, for lack of space we only introduce notions that are essential to show our results.

In particular, we report here the notion of *labeled granularity* which was proposed for the specification of a calendar algebra [BJW00, NWJ02].

Granularities are defined by grouping sets of instants into *granules*. For example, each granule of the granularity `day` specifies the set of instants included in a particular day. A label (or index) is used to refer to a particular granule. The whole set of time instants is called *time domain*, and for the purpose of this paper the domain can be an arbitrary infinite set with a total order relationship, $\leq$.

Definition. A *labeled granularity* is a pair $(\mathcal{L}, G)$, where $\mathcal{L}$ is a subset of the integers, and G is a mapping from $\mathcal{L}$ to the subsets of the time domain such that for each pair of integers i and j in $\mathcal{L}$ with $i < j$, if $G(i) \neq \emptyset$ and $G(j) \neq \emptyset$, then (1) each element in $G(i)$ is less than every element of $G(j)$, and (2) for each integer k in $\mathcal{L}$ with $i < k < j$, $G(k) \neq \emptyset$.

When $\mathcal{L}$ is exactly the integers, the granularity is called "full-integer labeled". When $\mathcal{L} = \mathbb{Z}^+$ we have the same notion of granularity as used in several applications (e.g., [BWJ02]). For example, following this labeling schema, if we assume to map `day`(1) to the subset of the time domain corresponding to January 1, 2001, `day`(32) would be mapped to February 1, 2001, `b-day`(6) to January 8, 2001 (the sixth business day), and `month`(15) to March 2002. The generalization to arbitrary label sets has been introduced mainly to facilitate conversion operations in the algebra, however our final goal is the conversion of a labeled granularity denoted by a calendar expression into a "positive-integer labeled" one denoted by a periodic formula.

Related to the notion of labeled granularity is the one of *label-aligned subgranularity*. Formally, G_1 is a label-aligned subgranularity of G_2 if the label set $\mathcal{L}_{G_1}$ of G_1 is a subset of the label set $\mathcal{L}_{G_2}$ of G_2 and for each i in $\mathcal{L}_{G_1}$ such that $G_1(i) \neq \emptyset$, we have $G_1(i) = G_2(i)$. Intuitively, G_1 has a subset of the granules of G_2 and those granules have the same label in the two granularities.

Several interesting relationships can be defined among granularities. The first of these is called *group into* and defines a partial order over the set of all granularities. If G and H are granularities, then G is said to *group into* H, denoted $G \trianglelefteq H$, if for each non-empty granule $H(j)$, there exists a (possibly infinite) set S of labels of G such that $H(j) = \bigcup_{i \in S} G(i)$. Intuitively, $G \trianglelefteq H$ means that each granule of H is a union of some granules of G. For example, `day` $\trianglelefteq$ `week` since a week is composed of 7 days and `day` $\trianglelefteq$ `b-day` since each business day is a day.

Granularities are said to be bounded when $\mathcal{L}$ has a first or last element or when $G(i) = \emptyset$ for some $i \in \mathcal{L}$. We assume the existence of an unbounded bottom granularity, denoted by $G_\perp$ which is full-integer labeled and groups into every other granularity in the system.

We also say that G *partitions* H if $G \trianglelefteq H$ and there are no granules of G other than those included in granules of H. For example, both `day` and `b-day` group into `b-week`, but `day` does not partition `b-week`, while `b-day` does.

Since we are particularly interested in granularities which can be expressed as periodic repetitions of granules of other granularities (in particular a bottom granularity), we formally define the following relationship[1]

Definition. A labeled granularity G *groups periodically into* a labeled granularity H if G groups into H and there exist positive integers N and P such that (1) for each label i of H, $i + N$ is a label of H unless $i + N$ is greater than the greatest label of H, and (2) for each label i of H, if $H(i) = \bigcup_{r=0}^{k} G(j_r)$ and $H(i + N)$ is a non empty granule of H then $H(i + N) = \bigcup_{r=0}^{k} G(j_r + P)$, and (3) if $H(s)$ is the first non-empty granule in H (if exists), then $H(s + N)$ is non empty.

Intuitively, condition (1) says that labels of H have N as a periodic pattern (i.e., if $i \in \mathcal{L}_H$, then $i + \alpha \cdot N \in \mathcal{L}_H$ with $\alpha \in \mathbb{Z}$ if H is unbounded), condition (2) says that the granules of H repeat along the time domain by "shifting" each granule by P granules of G, and condition (3) simply says that there is at least one of these repetitions. Several examples will be shown in Section 3.

We call each pair P and N in Definition 2, a *period* and its associated *period label distance*. Note that period and period label distance are not unique. More precisely, we indicate with P_H^G the period of H in terms of G and with N_H^G the period label distance of H in terms of G; the form P_H and N_H is used when $G = G_\perp$.

In general, this relationship guarantees that granularity H can be finitely described (in terms of granules of G) providing the following information: (i) a value for P and N; (ii) the set $\overline{\mathcal{L}}$ of labels of H in one period of length P; (iii)

[1] This is simply an extension to labeled granularities of the analogous relation defined for "regular" granularities (see e.g., [BJW00]).

for each $j \in \overline{\mathcal{L}}$, the finite set S_j of labels of G, describing the composition of $H(j)$; (iv) the labels of first and last non-empty granules in H, if their values are not infinite.

If $\overline{\mathcal{L}}$ is the set of labels of H with values in $\{b, \ldots, b + N - 1\}$, and we assume H to be unbounded, the description of an arbitrary granule $j \in \mathcal{L}_H$ can be obtained by the following formula.

Given $j' = [(j-1) \bmod N] + 1$ and

$$k = \begin{cases} \left(\left\lfloor \frac{b-1}{N} \right\rfloor\right) \cdot N + j' & \text{if } \left(\left\lfloor \frac{b-1}{N} \right\rfloor\right) \cdot N + j' \geq b \\ \\ \left(\left\lfloor \frac{b-1}{N} \right\rfloor + 1\right) \cdot N + j' & \text{otherwise} \end{cases}$$

we have

$$H(j) = \bigcup_{i \in S_k} G\left(P_H^G * \left\lfloor \frac{j-1}{N} \right\rfloor + i - P_H^G * \left\lfloor \frac{k-1}{N} \right\rfloor\right).$$

3. Deriving periodicity from algebraic operations

We now briefly illustrate each calendar algebra operation, identifying a pair of values P and N that characterize the periodicity of the granularity resulting from its application. Since we assume to start from the bottom granularity $G_\perp$, which is trivially periodic in terms of itself, these results applied recursively to a calendar expression, allow us to characterize the periodicity of the resulting granularity in terms of $G_\perp$.

With respect to the algebra defined in [NWJ02] we need to limit the *altering tick* operation to finite values (no ∞ in its parameters), and we exclude the *subset* operation if not simply applied as the last operation in an algebraic expression.

In the following, symbols G', G, G_1, and G_2 stand for granularities. G, G_1 and G_2 are usually operand granularities, while G' is the resulting granularity; Symbols $\mathcal{L}_{G'}$, $\mathcal{L}_G$, $\mathcal{L}_{G_1}$, and $\mathcal{L}_{G_2}$ stand for the label set associated to G', G, G_1, and G_2, respectively; Symbols $P_{G'}$, P, P_{G_1}, and P_{G_2} stand for periods of G', G, G_1, and G_2 respectively; $N_{G'}$, N_G,N_{G_1}, and N_{G_2} are the period label distances of G', G, G_1, and G_2, respectively.

3.1. The grouping operation

Let G be a full-integer labeled granularity, and m a positive integer. The grouping operation $\mathit{Group}_m(G)$ gives the granularity G' such that for each integer i,

$$G'(i) = \bigcup_{j=(i-1)\cdot m+1}^{i \cdot m} G(j).$$

For example, given granularity $\texttt{day}$, $\texttt{week}$ can be generated by $\texttt{week} = \mathit{Group}_7(\texttt{day})$ if we assume that $\texttt{day}(1)$ corresponds to Monday, i.e., the first day of a week. Note that G partitions G', and G' is also a full-integer labeled granularity.

Proposition 1 *If $G' = Group_m(G)$, then $P_{G'} = m \cdot P_G$ and $N_{G'} = N_G$*

Example 1 *Figure 1 shows an example of the grouping operation: granularity $\texttt{day}$ is the bottom granularity and its period is 1; $\texttt{week}$ is defined as $Group_7(\texttt{day})$, so it has a period of 7 days.*

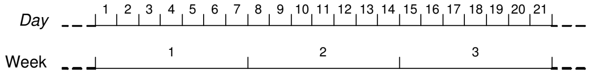

Figure 1. *Group operation* example

3.2. The altering-tick operation

Let G_1, G_2 be full-integer labeled granularities, and l, k, m integers, where G_2 partitions G_1, and $1 \leq l \leq m$. The altering-tick operation $Alter^m_{l,k}(G_2, G_1)$ generates a new granularity by periodically expanding or shrinking granules of G_1 in terms of granules of G_2. Since G_2 partitions G_1, each granule of G_1 consists of some contiguous granules of G_2. The granules of G_1 can be partitioned into m-granule groups such that $G_1(1)$ to $G_1(m)$ are in one group, $G_1(m+1)$ to $G_1(2m)$ are in the following group, and so on. The goal of the altering-tick operation is to modify the granules of G_1 so that the l-th granule of every aforementioned group will have $|k|$ additional (or fewer when $k < 0$) granules of G_2. For example, if G_1 represents 30-day groups (i.e., $G_1 = \mathit{Group}_{30}(\texttt{day})$) and we want to add a day to the third month every 12 month (i.e., to make March to have 31 days), we may perform $Alter^{12}_{3,1}(\texttt{day}, G_1)$.

Specifically, for all $i = l + m \cdot n$, where n is an integer, $G_1(i)$ denotes the granule to be shrunk or expanded. The granules of G_1 are split into two parts at $G_1(0)$. When $i > 0$, $G_1(i)$ expands (or shrinks) by taking in (or pushing out) later granules of G_2, and the effect is propagated to later granules of G_1. On the contrary, when $i \leq 0$, $G_1(i)$ expands (or shrinks) by taking in (or pushing out) earlier granules of G_2, and the effect is propagated to earlier granules of G_1.

The altering-tick operation can be formally described as follows. For each integer i such that $G_1(i) \neq \emptyset$, let b_i and t_i be the integers such that $G_1(i) = \cup_{j=b_i}^{t_i} G_2(j)$. (The integers b_i and t_i exist because G_2 partitions G_1.) Then $G' = \mathit{Alter}^m_{l,k}(G_2, G_1)$ is the granularity such that for each integer i, let $G'(i) = \emptyset$ if $G_1(i) = \emptyset$, and otherwise let

$G'(i) = \bigcup_{j=b'_i}^{t'_i} G_2(j)$ where:

$$b'_i = \begin{cases} b_i + (h-1) \cdot k, & \text{if } i = (h-1) \cdot m + l \\ b_i + h \cdot k, & \text{otherwise} \end{cases}$$

and $t'_i = t_i + h \cdot k$, having $h = \lfloor \frac{i-l}{m} \rfloor + 1$.

The original definition of altering-tick given in [NWJ02] as reported above, has the following problems when an arbitrary negative value for k is used: (1) It allows the definition of a G' that is not a full integer labeled granularity and (2) It allows the definition of a G' that does not even satisfy the definition of granularity. After considering several options, we decided that a reasonable restriction that avoids these undesired behaviors is imposing

$$k > -(mindist(G1, 2, G2) - 1)$$

where $mindist()$ is formally defined in [BJW00]. Intuitively, $mindist(G1, 2, G2)$ represents the minimum distance (in terms of granules of $G2$) between two consecutive granules of $G1$.

Proposition 2 *If $G' = Alter^m_{l,k}(G_2, G_1)$, then: $P_{G'} = m \cdot P_{G_1} \cdot P_{G_2} \cdot N_{G_2} + k \cdot N_{G_1} \cdot (P_{G_2})^2$ and $N_{G'} = m \cdot N_{G_1} \cdot N_{G_2} \cdot P_{G_2}$.*

Example 2 *Figure 2 shows an example of the altering tick operation: day is the bottom granularity; $G_1 = Group_3(Day)$ and G' is defined as $Alter^2_{2,-1}(Day, G_1)$. The period of G' is 5 and $N_{G'} = 2$.*

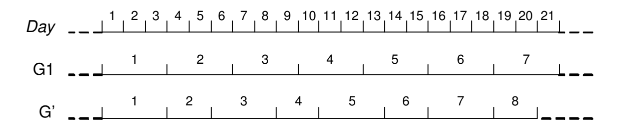

Figure 2. *Alter-tick* operation example

3.3. Shifting operation

Let G be a full-integer labeled granularity, and m an integer. The shifting operation generates a new granularity $G' = \textit{Shift}_m(G)$ such that for each integer i, $G'(i) = G(i+m)$. Note that G' is full-integer labeled.

Proposition 3 *If $G' = Shift_m(G_1)$, then $P_{G'} = P_{G_1}$ and $N_{G'} = N_{G_1}$.*

3.4. Combining operation

Let G_1 and G_2 be granularities with label sets $\mathcal{L}_{G_1}$ and $\mathcal{L}_{G_2}$ respectively. The combining operation $\textit{Combine}(G_1, G_2)$ generates a new granularity G' by combining all the granules of G_2 that are included in one granule of G_1 into one granule of G'. More formally, for each $i \in \mathcal{L}_{G_1}$, let $s(i) = \emptyset$ if $G_1(i) = \emptyset$, and otherwise let $s(i) = \{j \in \mathcal{L}_{G_2} | \emptyset \neq G_2(j) \subseteq G_1(i)\}$. Then $G' = \textit{Combine}(G_1, G_2)$ is the granularity with the label set $\mathcal{L}_{G'} = \{i \in \mathcal{L}_{G_1} | s(i) \neq \emptyset\}$ such that for each i in $\mathcal{L}_{G'}$, $G'(i) = \bigcup_{j \in s(i)} G_2(j)$.

As an example, given granularities `b-day` and `month`, the granularity for business months can be generated by

$$\texttt{b-month} = \textit{Combine}(\texttt{month}, \texttt{b-day}).$$

Proposition 4 *If $G' = Combine(G_1, G_2)$, then: $P_{G'} = lcm(P_{G_1}, P_{G_2})$ and $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}}$*

Example 3 *Figure 3 shows an example of the combining operation: $G' = Combine(Week, G_2)$. Day is the bottom granulairty, the period of week is 7 and the period of G_2 is 3, therefore the period of G' is 21.*

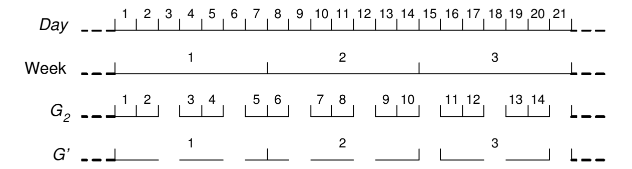

Figure 3. *Combining* operation example

Anchored grouping operation

Let G_1 and G_2 be granularities with label sets $\mathcal{L}_{G_1}$ and $\mathcal{L}_{G_2}$ respectively, where G_2 is a label-aligned subgranularity of G_1, and G_1 is a full-integer labeled granularity. The anchored grouping operation $\textit{Anchored-group}(G_1, G_2)$ generates a new granularity G' by combining all the granules of G_1 that are between two granules of G_2 into one granule of G'. More formally, $G' = \textit{Anchored-group}(G_1, G_2)$ is the granularity with the label set $\mathcal{L}_{G'} = \mathcal{L}_{G_2}$ such that for each $i \in \mathcal{L}_{G'}$, $G'(i) = \cup_{j=i}^{i'-1} G_1(j)$, where i' is the next label of G_2 after i.

Proposition 5 *If $G' = \textit{Anchored-group}(G_1, G_2)$, then $P_{G'} = lcm(P_{G_1}, P_{G_2})$ and $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) \cdot N_{G_2}}{P_{G_2}}$*

Example 4 *Figure 4 shows an example of the anchored grouping operation: the `USweek` (i.e., a week starting with a Sunday) is defined by the operation `anchored-group(day, Sunday)`. Using `day` as the bottom granularity, the period of `USWeek` is 7.*

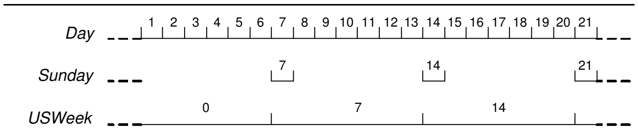

Figure 4. *Anchored grouping* operation example

3.5. Subset operation

The subset operation is designed to generate a new granularity by selecting an interval of granules from another granularity. Let G be a granularity with label set $\mathcal{L}$, and m, n integers such that $m \leq n$. The subset operation $G' =$ *Subset*$^n_m(G)$ generates a new granularity $G' =$ *Subset*$^n_m(G)$ with the label set $\mathcal{L}_{G'} = \{i \in \mathcal{L} \mid m \leq i \leq n\}$, and for each $i \in \mathcal{L}_{G'}$, $G'(i) = G(i)$. Note that G' is a label-aligned subgranularity of G, and G' is not a full-integer labeled granularity even if G is. We also allow the extensions of setting $m = -\infty$ or $n = \infty$ with semantics properly extended.

This operation returns a finite granularity. It may still be periodic, with same period as its argument granularity, if it includes granules that cover exactly a multiple of its period, provided that we introduce bounds for the description of the periodic granularity. Clearly, these bounds are determined by the values of m and n when different from $-\infty$ and $+\infty$, respectively. However, in general, either the operation will lead to a finite non-periodic granularity (whose granules are all explicitly described), or to a finite semi-periodic granularity [BJW00], where the periodic part has the same period as its argument granularity.

3.6. Selecting operations

There are three selecting operations: *select-down, select-up* and *select-by-intersect*. To facilitate the description of these operations, the $\Delta^l_k(S)$ notation is used. Intuitively, if S is a set of integers, $\Delta^l_k(S)$ selects l elements starting from the k-th one.

Select-down operation. For each granule $G_2(i)$, there exits a set of granules of G_1 that is contained in $G_2(i)$. The operation *Select-down*$^l_k(G_1, G_2)$, where $k \neq 0$ and $l > 0$ are integers, selects granules of G_1 by using $\Delta^l_k(\cdot)$ on each set of granules (actually their labels) of G_1 that are contained in one granule of G_2. More formally, $G' =$ *Select-down*$^l_k(G_1, G_2)$ is the granularity with the label set

$$\mathcal{L}_{G'} = \cup_{i \in \mathcal{L}_{G_2}} \Delta^l_k(\{j \in \mathcal{L}_{G_1} \mid \emptyset \neq G_1(j) \subseteq G_2(i)\}),$$

and for each $i \in \mathcal{L}_{G'}$, $G'(i) = G_1(i)$.

Select-up and Select-by-intersect operations. Two analogous operations can be defined. The *Select-up*(G_1, G_2) operation generates a new granularity G' by selecting the granules of G_1 that contain one or more granules of G_2. The *Select-by-intersect*$^l_k(G_1, G_2)$ operation, where $k \neq 0$ and $l > 0$ are integers, selects granules of G_1 by applying for each granule $G_2(i)$ the $\Delta^l_k(\cdot)$ operator to the set of labels of $\mathcal{L}_{G_1}$ such that the corresponding granules intersect $G_2(i)$. In both cases, G' is a label-aligned subgranularity of G_1.

Proposition 6 *If $G' =$ Select-down$^l_k(G_1, G_2)$ or $G' =$ Select-up(G_1, G_2) or $G' =$ Select-by-intersect(G_1, G_2) then $P_{G'} = lcm(P_{G_1}, P_{G_2})$ and $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}}$.*

Example 5 *Consider a calendar with* `day` *as the bottom granularity.* `Sunday` *could be defined as Select-down$^1_7$(*`day`*,* `week`*). Hence, the period of the granularity* `Sunday` *is 7.*

3.7. Set operations

The set operations, *union*, *intersection*, and *difference*, can be defined on G_1 and G_2 only if there exists a granularity H such that G_1 and G_2 are both label-aligned subgranularities of H.

Union. The union operation $G_1 \cup G_2$ generates a new granularity G' by collecting all the granules from both G_1 and G_2. More formally, $G' = G_1 \cup G_2$ is the granularity with the label set $\mathcal{L}_{G'} = \mathcal{L}_{G_1} \cup \mathcal{L}_{G_2}$, and for each $i \in \mathcal{L}_{G'}$,

$$G'(i) = \begin{cases} G_1(i), & i \in \mathcal{L}_{G_1}, \\ G_2(i), & i \in \mathcal{L}_{G_2} - \mathcal{L}_{G_1}. \end{cases}$$

Note that G_1 and G_2 are label-aligned subgranularities of G'.

Example 6 *Given granularities* `Saturday` *and* `Sunday`*, the granularity* `WeekendDay` *can be generated by* `WeekendDay` = `Sunday` $\cup$ `Saturday`.

The operations intersection $\cap$ and difference $(-)$ are defined similarly.

Since a set operation is valid if the granularities used as argument are both labeled aligned granularity of another granularity, the following property is used.

Proposition 7 *If G is a labeled aligned subgranularity of H, then $\frac{N_G}{P_G} = \frac{N_H}{P_H}$.*

Proposition 8 *If $G' = G_1 \cup G_2$ or $G' = G_1 \cap G_2$ or $G' = G_1 - G_2$ then $P_{G'} = lcm(P_{G_1}, P_{G_2})$ and $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_2}}{P_{G_2}}$.*

4. Completing the periodical characterization

In Section 3 we have derived for each operation the values P and N characterizing the periodicity of the resulting granularity in terms of one of the operand granularities. In order to fully characterize the granularity we still have to provide a description of the granules in one of the periods, assuming, of course, that we have one for the operand granularities. Moreover, we should identify possible lower and upper bounds.

For what concern the bounds, we know that each operation, except *subset*, results in an unbounded granularity if the granularities used as arguments are unbounded. From this consideration and since we assume that the bottom granularity is unbounded, all the granularities that could be generated with the calendar operation (excluding the subset operation) are unbounded. Since we assume that the subset operation could be only used as the last operation in an algebraic expression, we can assure that for each operation, except the subset, the resulting granularity is unbounded. For the *subset* operation it has been indicated in the previous section how to calculate the lower and upper bounds.

Since we do not have to deal with bounds and since the period can be easily obtained, in order to fully characterize a periodical granularity we only need the explicit representation of the granules in one of its periods.

4.1. Deriving the explicit granules in one period

Observe that our algebra operations are always defined in two step, first the label set, and second the granules for each label. In order to derive the explicit granules in one period, we first need to identify the set of labels $\overline{\mathcal{L}}_{G'}$ for the granules of the resulting granularity in one period of G', and then compute the value of each of them according to the operation definition.

In order to check the relationship among explicit granules as directed by the algebra operation definitions, for each granularity G, we define $\hat{\mathcal{L}}_G$ as the set of labels corresponding to granules of G which contain at least one of $G_\perp(1), \ldots, G_\perp(P_G)$.[2] $\overline{\mathcal{L}}_G$ is equal to $\hat{\mathcal{L}}_G$ if $G_\perp(0) \notin G(min(\hat{\mathcal{L}}_G))$, otherwise $\overline{\mathcal{L}}_G = \hat{\mathcal{L}}_G - \{max(\hat{\mathcal{L}}_G)\}$. To compute $\overline{\mathcal{L}}_{G'}$ we still need to extend both explicit granules and labels of each operand granularity to the period of G', and then use the operation definitions to derive $\hat{\mathcal{L}}_{G'}$ and then $\overline{\mathcal{L}}_{G'}$. Given a granularity G with period P_G, the corresponding set of labels $\hat{\mathcal{L}}_G$ can be extended to a period P by considering P instead of P_G in the definition of $\hat{\mathcal{L}}$. We denote the resulting set with $\hat{\mathcal{L}}_G^P$.

The idea is to start with $\hat{\mathcal{L}}_{G_\perp} = \{1\}$, and then show for each operation how $\hat{\mathcal{L}}$ of the resulting granularity G' is calculated given the $\hat{\mathcal{L}}$ value (possibly extended to the common period $P_{G'}$) of the operand granularities.

If $G' = Shift_m(G_1)$ then $\hat{\mathcal{L}}_{G'} = \{i \in \mathbb{Z} | i + m \in \hat{\mathcal{L}}_{G_1}\}$
If $G' = Group_m(G_1)$ then
$\quad \hat{\mathcal{L}}_{G'} = \{min(\hat{\mathcal{L}}_{G_1}), \ldots, min(\hat{\mathcal{L}}_{G_1}) + N_{G'}\}$
If $G' = AlterTick^m_{l,k}(G_2, G_1)$ then
$\quad \hat{\mathcal{L}}_{G'} = \{min(\hat{\mathcal{L}}_{G_2}), \ldots, min(\hat{\mathcal{L}}_{G_2}) + N_{G'}\}$
If $G' = Combining(G_1, G_2)$ then
$\quad \forall i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}\ \bar{s}(i) = \{j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} \,|\, \emptyset \neq G_2(j) \subseteq G_1(i)\}$
$\quad$ and $\hat{\mathcal{L}}_{G'} = \{i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \bar{s}(i) \neq \emptyset\}$
If $G' = select\_down^l_k(G_1, G_2)$ then
$\quad \hat{\mathcal{L}}_{G'} = \bigcup_{i \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}} \Delta^l_k \left(\left\{j \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \emptyset \neq G_1(j) \subseteq G_2(i)\right\}\right)$
If $G' = select\_up(G_1, G_2)$ then
$\quad \hat{\mathcal{L}}_{G'} = \left\{j \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \exists j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} (\emptyset \neq G_2(j) \subseteq G_1(i))\right\}$
If $G' = select\_by\_intersect(G_1, G_2)$ then
$\quad \hat{\mathcal{L}}_{G'} = \bigcup_{i \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}} \Delta^l_k \left(\left\{j \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \emptyset \neq G_1(j) \cap G_2(i)\right\}\right)$
If $G' = G_1 \cup G_2$ then
$\quad \hat{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G_1}^{P_{G'}} \cup \hat{\mathcal{L}}_{G_2}^{P_{G'}}$, and similarly for $\cap$ and $-$.

Example 7 *Consider a calendar having* day *as the bottom granularity; let* week *be defined with* $P_{\mathsf{week}} = 7$, $N_{\mathsf{week}} = 1$ *and* $\hat{\mathcal{L}}_{\mathsf{week}} = \{1\}$. *We define the granularity including all the Saturdays and Sundays (called* wday*) as* $\mathsf{wday} = select\_down^2_6(\mathsf{day}, \mathsf{week})$. *By Proposition 6,* $P_{\mathsf{wday}} = 7$ *and* $N_{\mathsf{wday}} = 7$. *In order to compute* $\hat{\mathcal{L}}_{\mathsf{wday}}$ *we first need to calculate* $\hat{\mathcal{L}}_{\mathsf{day}}^{P_{wday}} = \{1, 2, 3, 4, 5, 6, 7\}$, *and* $\hat{\mathcal{L}}_{\mathsf{week}}^{P_{wday}} = \{1\}$. *Hence, by definition of the* combining *operation* $\hat{\mathcal{L}}_{\mathsf{wday}} = \Delta^2_6(j \in \hat{\mathcal{L}}_{\mathsf{day}}^{P_{wday}} \,|\, \emptyset \neq \mathsf{day}(j) \subseteq \mathsf{week}(1)) = \{6, 7\}$.

Since $\overline{\mathcal{L}}_{G'}$ is trivially derived from $\hat{\mathcal{L}}_{G'}$, we can calculate the explicit granules in one period by applying the formulas specific to the operation for each label in $\overline{\mathcal{L}}_{G'}$. Referring to Example 7, $\overline{\mathcal{L}}_{\mathsf{wday}} = \hat{\mathcal{L}}_{\mathsf{wday}} = \{6, 7\}$, and using the definition of the *select_down* operation we derive $\mathsf{wday}(6) = \mathsf{day}(6)$ and $\mathsf{wday}(7) = \mathsf{day}(7)$, which together with the values of P_{week} and N_{week} fully characterize week.

If we need to characterize all granularities as periodic expressions in terms of the bottom granularity, we can simply recursively apply the steps we have shown for each algebraic operation.

Example 8 *Continuing with Example 7, suppose we define* $\mathsf{weekend}$ *as* combine*(*$\mathsf{week}, \mathsf{wday}$*). By Proposition 4* $P_{\mathsf{weekend}} = 7$ *and* $N_{\mathsf{weekend}} = 1$*; moreover from the formulas to derive* $\hat{\mathcal{L}}$*:* $\hat{\mathcal{L}}_{\mathsf{weekend}} = \overline{\mathcal{L}}_{\mathsf{weekend}} = \{1\}$. *By definition of the* combining *operation,* $s(1) = \{6, 7\}$, *and hence* $\mathsf{weekend}(1) = \bigcup_{j \in \{6,7\}} \mathsf{wday}(j)$. *Since* $\mathsf{wday}(6) = \mathsf{day}(6)$ *and* $\mathsf{wday}(7) = \mathsf{day}(7)$, *then* $\mathsf{weekend}(1) = \mathsf{day}(6) \cup \mathsf{day}(7)$.

[2] Note that the cardinality of $\hat{\mathcal{L}}_G$ will be equal to the number of granules in one period or have one extra label (in the case a granule includes both $G_\perp(0)$ and $G_\perp(1)$).

4.2. Computing arbitrary values from the periodic representation

Given $\overline{\mathcal{L}}_{G'}$, the explicit granules, and the values of $P_{G'}$ and $N_{G'}$ for the granularity G' derived by an algebraic operation where G is the operand granularity which periodically groups into G', we can easily compute the composition of an arbitrary granule $G'(j)$ with $j \in \mathcal{L}_{G'}$ in terms of granules of G using the formula given at the end of Section 2.

Example 9 *Referring to Example 7, we know that* $\overline{\mathcal{L}}_{\mathsf{wday}} = \{6, 7\}$ *and* $\mathsf{wday}(6) = \mathsf{day}(6)$ *and* $\mathsf{wday}(7) = \mathsf{day}(7)$. $\mathsf{wday}(14)$ *can be computed using the formula observing that* $j = 14$, $b = min(\overline{\mathcal{L}}_{\mathsf{wday}}) = 6$, $N = P_H^G = 7$*, and* $S_6 = \{6\}$ *and* $S_7 = \{7\}$*. Then we derive* $k = j' = 7$*, and hence* $\mathsf{wday}(14) = \mathsf{day}(14)$.

4.3. Reducing the periodic representation to positive-integer labeled granularities

Each granularity obtained from a calendar expression as explained above can be easily converted into a full- or positive-integer labeled granularity in order to satisfy the definition used in some of the existing applications (e.g., GSTP). The formal definition of this process is described in [BJW00] and is not reported here for lack of space.

Example 10 *Referring to Example 7, the granules with the labels* 6 *and* 7 *in* $\overline{\mathcal{L}}_{\mathsf{wday}}$ *are relabeled with* 1 *and* 2*, respectively. While the value of the period remains the same, the value of* N_{wday} *is changed into* 2*, the number of granules in each period. It is easily seen that the formula provided to compute arbitrary values is still valid.*

5. Conclusions

We have shown how, given a calendar algebra expression, we can derive an equivalent granularity characterization in terms of periodic sets of granules of the bottom granularity. The periods and period label distances we obtain are not always minimal. We are investigating algorithms to reduce the period value or even to find the minimum values. The optimal period values have more general applications; for example it would be very useful even to further reduce the representations given by users that directly define their granularities in terms of periodic sets, as it currently happens for the GSTP system [BWJ02].

We are also working on XML versions of algebraic and periodical representations, to enable a rich set of time granularity web services that could be useful for many applications handling time-dependent data.

References

[BJW00] C. Bettini, S. Jajodia, and X. Wang. *Time Granularities in Databases, Temporal Reasoning, and Data Mining*. Springer, 2000.

[BD00] C. Bettini, R. De Sibi. Symbolic Representation of User-Defined Time Granularities, *Annals of Mathematics and Artificial Intelligence*, 30(1-4):53-92, 2000.

[BWJ02] C. Bettini, X. Wang, S. Jajodia, Solving Multi-Granularity constraint networks, *Artificial Intelligence*, 140(1-2):107–152, 2002.

[EJS98] O. Etzion, S. Jajodia, and S. Sripada (Eds.) Temporal Databases: Research and Practice. Springer-Verlag Heidelberg, Lecture Notes in Computer Science 1380, 1998.

[LMF86] B. Leban, D. Mcdonald, and D. Foster, A representation for collections of temporal intervals, in *Proc. of the American National Conference on Artificial Intelligence*, pp. 367–371, AAAI Press, 1986.

[Nie92] M. Niezette and J. Stevenne, An efficient symbolic representation of periodic time, in *Proc. of International Conference on Information and Knowledge Management*, pp. 161–168, ACM Press, 1992.

[NWJ02] P. Ning, X. Wang, S. Jajodia. An Algebraic Representation of Calendars. *Annals of Mathematics and Artificial Intelligence* 36(1-2): 5–38, 2002.

[Ter03] Paolo Terenziani, Symbolic User-Defined Periodicity in Temporal Relational Databases. *IEEE Trans. on Knowledge and Data Engineering*, 15(2): 489–509, 2003.