Università degli Studi di Milano
Dipartimento di Informatica e Comunicazione

Rapporto interno N. 04-05

Minimal Periodical Representations of Calendar Expressions

Claudio Bettini, Sergio Mascetti

Minimal Periodical Representations of Calendar Expressions

Claudio Bettini and Sergio Mascetti

DICO Technical Report N. 04-05

Abstract. The formalization of time granularity, the study of computational methods to manipulate granularities and their applications have been the subject of much research in the recent years. This paper aims at filling the gap between symbolic representations defined thrugh calendar algebras and mathematical representations defined through periodical sets of instants. Considering a specific calendar algebra, the paper shows how each algebraic operator changes the periodical structure of the granularities given as operands. This leads to a conversion procedure between calendar expressions and periodical representations of granularities. From a practical point of view, our results allow users to specify granularities using the algebra (which is more natural) and to process them as periodical sets (as most current computational methods do). The paper also shows experimental evidence that the provided mapping is in most cases optimal and presents an efficient post-processing method for those cases where optimality must be always guaranteed.

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 [BJW00], where integer intervals are often used internally as timestamps, and temporal constraint 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 work, 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 this work 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. 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.

2 Formal notions of time granularities

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.

2.1 Time Granularities

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]; we will show later how any *labeled granularity* can be reduced to a more standard notion of granularity, like the one used in [BWJ02].

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

When $\mathcal{L}_G$ is exactly the integers, the granularity is called "full-integer labeled". When $\mathcal{L}_G = \mathbb{Z}^+$ 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.

2.2 Granularity Relationships

Some interesting relationships between granularities follows. The definitions are extended from the ones presented in [BJW00] to cover the notion of labeled granularity.

Definition 2. *If G and H are labeled 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.

Definition 3. *If G and H are labeled granularities, then G is said to be* finer than *H, denoted $G \preceq H$, if for each non-empty granule $G(i)$, there exists a granule $H(j)$ such that $G(i) \subseteq H(j)$.*

For example `business-day` is finer than `day`, and also finer than `week`.

We also say that G *partitions* H if $G \trianglelefteq H$ and $G \preceq H$. Intuitively 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` (business week, i.e., the business day in a week), but `day` does not partition `b-week`, while `b-day` does.

Definition 4. *A labeled granularity G_1 is a* label-aligned subgranularity *of a labeled granularity 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.

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

2.3 Granularity Conversions

When dealing with granularities, we often need to determine the granule (if any) of a granularity H that covers a given granule z of another granularity G. For example, we may wish to find the month (an interval of the absolute time) that includes a given week (another interval of the absolute time).

This transformation is obtained with the *up* operation. Formally, for each label $z \in \mathcal{L}_G$, $\lceil z \rceil_G^H$ is undefined if $\nexists z' \in \mathcal{L}_H$ s.t. $G(z) \subseteq H(z')$; otherwise, $\lceil z \rceil_G^H = z'$, where z' is the unique index value such that $G(z) \subseteq H(z')$. The uniqueness of z' is guaranteed by the monotonicity of granularities. As an example, $\lceil z \rceil_{\texttt{second}}^{\texttt{month}}$ gives the month that includes the second z. Note that while $\lceil z \rceil_{\texttt{second}}^{\texttt{month}}$ is always defined, $\lceil z \rceil_{\texttt{week}}^{\texttt{month}}$ is undefined if week z falls between two months. Note that if $G \preceq H$, then the function $\lceil z \rceil_G^H$ is defined for each index value z. For example, since $\texttt{day} \preceq \texttt{week}$, $\lceil z \rceil_{\texttt{day}}^{\texttt{week}}$ is always defined, i.e., for each day we can find the week that contains it. The notation $\lceil z \rceil^H$ is used when the source granularity can be left implicit (e.g., when we are dealing with a fixed set of granularities having a distinguished bottom granularity).

Another direction of the above transformation is the *down* operation: Let G and H be granularities such that $G \trianglelefteq H$, and z an integer. Define $\lfloor z \rfloor_G^H$ as the set S of labels of granules of G such that $\bigcup_{j \in S} G(j) = H(z)$.[1] This function is useful for finding, e.g., all the days in a month.

2.4 The Periodical Granules Representation

A central issue in temporal reasoning is the possibility of finitely representing infinite granularities. The definition of granularity provided above is general and expressive but it may be impossible to provide a finite representation of some of the granularities. Even labels (i.e., a subset of the integers) do not necessary have a finite representation.

A solution has been first proposed in [BJW00]. The idea is that most of the commonly used granularities present a periodical behavior; it means that there is a certain pattern that repeats periodically. This feature has been exploited to provide a method for finitely describe granularities. The formal definition is based on the *periodically groups into* relationship.

Definition 5. *A labeled granularity G groups periodically into a labeled granularity H ($G \,\bar{\trianglelefteq}\, H$) if $G \trianglelefteq H$ 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

[1] This definition is different from the one given in [BJW00] since non contiguous granules of G are considered

(3) if $H(s)$ is the first non-empty granule in H (if exists), then $H(s+N)$ is non empty.

The *groups periodically into* relationship is a special case of the group into characterized by a periodic repetition of the "grouping pattern" of granules of G into granules of H. Its definition may appear complicated but it is actually quite simple. Since G groups into H, any granule $H(i)$ is the union of some granules of G; for instance assume it is the union of the granules $G(a_1), G(a_2), \ldots, G(a_k)$. Condition (1) ensures that the label $i+N$ exists (if it not greater than the greatest label of H) while condition (2) ensures that, if $H+N$ is not empty, then it is the union of $G(a_1+P), G(a_2+P), \ldots, G(a_k+P)$. Condition (3) simply says that there is at least one of these repetition.

We call each pair P and N in Definition 5, a *period* and its associated *period label distance*. We also indicate with R the number of granules of G corresponding to each groups of P consecutive granules of $\perp$. More formally R is equal to the number of labels of G greater or equal than i and smaller than $i+N$ where i is an arbitrary label of G. Note that R is not affected by the value of i.

The period and the 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 $G = \perp$.

Note that *the* period is an integer value. For simplicity we also indicate with *one* period of a granularity G a set of R consecutive granules of G.

In general, the periodically-groups-into 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 $\mathcal{L}^P$ of labels of H in one period of H; (iii) for each $j \in \mathcal{L}^P$, 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 a granularity H can be represented as a periodic set of granules of a granularity G, then there exists an infinite number of pairs (P_H^G, N_H^G) each one satisfying the periodically-groups-into relation. If a pair (P, N) satisfies the relation, it can be proved that for each $\alpha \in \mathbb{N}^+$ the pair $(\alpha P, \alpha N)$ satisfies the periodically-groups-into relation. However there exists a pair (P, N) such that P is the smallest period value in all pairs; a representation adopting that value for the period is called *minimal*.

If H is fully characterized in terms of G, it is possible to derive the composition, in terms of G, of any granule of H. Indeed, if $\mathcal{L}^P$ 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 $H(j)$ 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

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

Example 1. Figure 1 shows granularities G and H where $G \trianglelefteq H$. H is fully characterized in terms of G since we know that $P_H^G = 4$, $N_H^G = 3$, $\mathcal{L}_H^P = \{22, 24\}$, $S(22) = \{27\}$ and $S(24) = \{29, 30\}$. If we want to compute the composition of the granule $H(15)$, we apply the formula presented above: $j' = 3$; $k = 24$. Then $H(15) = G(4 \cdot 4 + 29 - 4 \cdot 7) \cup G(4 \cdot 4 + 30 - 4 \cdot 7) = G(17) \cup G(18)$.

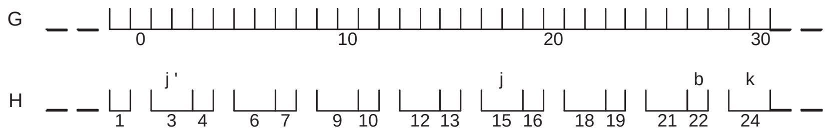

Fig. 1. *Periodically groups into* example

3 Calendar Algebra

Several high-level symbolic formalisms have been proposed to represent granularities [LMF86], [Nie92]. Recently some extensions have been presented (e.g. [BD00]) and an alternative formalism (e.g. [NWJ02]) have been proposed. The basic idea is that granularities generally are not isolated, but closely related and some algebraic operations can be introduced to capture these relationships. Those operators can be applied to any existing granularity (not only to the bottom granularity) allowing an incremental definition of granularities. For many applications this approach is more immediate and user friendly than indicating the granules of bottom that compose the granules in one period of the granularity to be defined.

In this work we consider the formalism proposed by Wang et al. called *Calendar Algebra*. In this approach a set of algebraic operations is defined; each operation generates a new granularity by manipulating other granularities that have already been generated. The relationships between the operands and the resulting granularities are thus encoded in the operations. All granularities that are generated directly or indirectly from the same one (called the bottom granularity) form a calendar, and these granularities are related to each other through the operations that define them. In practice, the choices for the bottom granularity include `day`, `hour`, `second`, `microsecond` and other granularities, depending on the accuracy required in each application context.

The calendar algebra consists of the following two kinds of operations: the *grouping-oriented operations* and the *granule-oriented operations*. The grouping-oriented operations group certain granules of a granularity together to form new granules in a new granularity, while the granule-oriented operations don't change the granules of a granularity, but rather make choices as to which granules should remain in the new granularity.

Certain calendar operations will only work on full-integer labeled granularities, while others will be more easily defined and implemented using more flexible labelings. The calendar algebra can be easily adapted to generate only full-integer labeled granularities as well as granularities with restricted use of integers as indices for granules. The idea is to use the subset operation and a "relabel" operation as the last steps to obtain the desired granularity. For example, for granularity systems where each granule has to be indexed by a positive integer with the first granule indexed by 1, the subset operation will select a subset of the granules in which the first granule exists, and the relabel operation will simply label the first granule as 1, the second as 2, and so on. We will come back to this issue later (see Section 4.10).

In the following we illustrate the calendar algebra operations presented in [NWJ02] together with some restriction introduced in [BMW04].

3.1 The Grouping-Oriented Operations

The grouping operation Let G be a full-integer labeled granularity, and m a positive integer. The grouping operation $Group_m(G)$ generates a new granularity G' by partitioning the granules of G into m-granule groups and making each group a granule of the resulting granularity. More precisely, $G' = Group_m(G)$ is the granularity such that for each integer i,

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

For example, given granularity `day`, `week` can be generated by `week` $= Group_7($`day`$)$ if we assume that `day`(1) corresponds to Monday, i.e., the first day of a week. Similarly, given `month`, `year` can be generated by `year` $= Group_{12}($`month`$)$ if we assume `month`(1) corresponds to a January. Note that G partitions G', and G' is also a full-integer labeled granularity.

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.,

$G_1 = Group_{30}(\texttt{day}))$ and we want to add a day to every 3-rd month (i.e., to make March to have 31 days), we may perform $Alter^{12}_{3,1}(\texttt{day}, G_1)$.

More 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 $G' = Alter^m_{l,k}(G_2, G_1)$ 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

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

$$t'_i = t_i + h \cdot k,$$

and

$$h = \lfloor \frac{i-l}{m} \rfloor + 1.$$

Example 2. Figure 2 shows an example of the `Alter` operation. Granularity G_1 is defined by $G_1 = Group_5(G_2)$ and granularity G' is defined by $G' = Alter^2_{2,-1}(G_2, G_1)$, which means shrinking the second one of every two granules of G_1 by one granule of G_2.

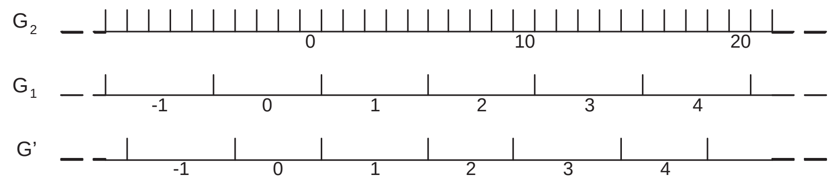

Fig. 2. *Altering-tick* operation example

Note that the grouping operation is a special case of the altering-tick operation. Indeed, $Group_m(G) = Alter^1_{1,m-1}(G, G)$, i.e., grouping every m granules together is the same as expanding every granule by $m-1$ granules. However, we keep grouping operation in the calendar algebra because of its conceptual and computational simplicity.

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 $G1$.

The shift operation Let G be a full-integer labeled granularity, and m an integer. The shifting operation $\mathit{Shift}_m(G)$ generates a new granularity G' by shifting the labels of G by m positions. More formally, $G' = \mathit{Shift}_m(G)$ is the granularity such that for each integer i, $G'(i) = G(i - m)$. Note that G' is also full-integer labeled.

The 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 $\mathit{Combine}(G_1, G_2)$ 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 $i \in \mathcal{L}_1$, 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 $G' = \mathit{Combine}(G_1, G_2)$ 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

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

The 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 $\mathit{Anchored\text{-}group}(G_1, G_2)$ 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, $G' = \mathit{Anchored\text{-}group}(G_1, G_2)$ 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.

Granularity G_2 is called the *anchor granularity* of G_1 in this operation. The granules of G_2 divide the granules of G_1 into groups, and each group is made a resulting granule by the anchored grouping operation.

For example, each academic year at a certain university begins on the last Monday in August, and ends on the day before the beginning of the next academic year. Then, the granularity corresponding to the academic years can be generated by

$$AcademicYear = \mathit{Anchored\text{-}group}(\mathtt{day}, \mathtt{lastMondayOfAugust}).$$

3.2 The Granule-Oriented Operations

The 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 $\mathcal{L}_G$, and m, n integers such that $m \leq n$. The subset operation $G' = \mathit{Subset}_m^n(G)$ generates a new granularity G' by taking all the granules of G whose labels are between m and n. More formally, $G' = \mathit{Subset}_m^n(G)$ is the granularity with the label set $\mathcal{L}_{G'} = \{i \in \mathcal{L}_G \mid m \leq i \leq n\}$, and for each $i \in \mathcal{L}_{G'}$,

$$G'(i) = G(i).$$

For example, given granularity `year`, all the years in the 20th century can be generated by

$$\mathtt{20CenturyYear} = \mathit{Subset}_{1900}^{1999}(\mathtt{year}).$$

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.

The selecting operations The selecting operations are all binary operations. They generate new granularities by selecting granules from the first operand in terms of their relationship with the granules of the second operand. The result is always a label-aligned subgranularity of the first operand granularity.

There are three selecting operations: *select-down*, *select-up* and *select-by-intersect*. To facilitate the description of these operations, we introduce a notation for subsets of a given set of integers. Suppose S is a set of n integers. Let $S = \{j_1, j_2, \ldots, j_n\}$, where $j_1 < j_2 < \ldots < j_n$. For each $i \leq 0$, let j_i be an arbitrary integer less than j_1, and for each $i > n$, let j_i be an arbitrary integer greater than j_n. Given two integers k and l, where $k \neq 0$ and $l > 0$, $\Delta_k^l(S)$ denotes the subset of S defined as follows:

$$\Delta^l_k(S) = \begin{cases} S \cap \{j_k, \ldots, j_{k+l-1}\}, & \text{if } k > 0, \\ S \cap \{j_{(n+k+2)-1}, \ldots, j_{(n+k+2)-l}\}, & \text{if } k < 0. \end{cases}$$

Therefore, $\Delta^l_k(S)$ consists of the l (or less than l if the range determined by k and l is out of S) integers in S starting from the k-th one from the beginning of the list counting forward (or the $|k|$-th one from the end of the list counting backward if $k < 0$). For example,

$$\Delta^2_3(\{1,2,3,4,5,6,7\}) = \{3,4\} \text{ and } \Delta^2_{-7}(\{1,2,3,4,5,6,7\}) = \{1\}.$$

Let G_1 and G_2 be granularities with label sets $\mathcal{L}_{G_1}$ and $\mathcal{L}_{G_2}$ respectively. In the following, we describe the selecting operations using the $\Delta^l_k(\cdot)$ operator.

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 $\textit{Select-down}^l_k(G_1, G_2)$, 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, $G' = \textit{Select-down}^l_k(G_1, G_2)$ 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 $i \in \mathcal{L}_{G'}$,

$$G'(i) = G_1(i).$$

For example, Thanksgiving days are the fourth Thursdays of all Novembers. If granularities `Thursday` and `November` are given, it can be generated by

$$\texttt{Thanksgiving} = \textit{Select-down}^1_4(\texttt{Thursday}, \texttt{November}).$$

Note that G' is a label-aligned subgranularity of G_1.

Select-up operation. The select-up operation $\textit{Select-up}(G_1, G_2)$ generates a new granularity G' by selecting the granules of G_1 that contain one or more granules of G_2. More formally, $G' = \textit{Select-up}(G_1, G_2)$ is the granularity with the label set

$$\mathcal{L}_{G'} = \{i \in \mathcal{L}_{G_1} | \exists j \in \mathcal{L}_{G_2}(\emptyset \neq G_2(j) \subseteq G_1(i)), \}$$

and for each $i \in \mathcal{L}_{G'}$,

$$G'(i) = G_1(i).$$

For example, given granularities `week` and `Thanksgiving`, the weeks that contain Thanksgiving days can be defined by

$$\texttt{ThanxWeek} = \textit{Select-up}(\texttt{week}, \texttt{Thanksgiving}).$$

Note that G' is a label-aligned subgranularity of G_1.

Select-by-intersect operation. For each granule $G_2(i)$, there may exist a set of granules of G_1, each intersecting $G_2(i)$. The operation $\textit{Select-by-intersect}^l_k(G_1, G_2)$, where $k \neq 0$ and $l > 0$ are integers, selects granules of G_1 by applying $\Delta_k^l(\cdot)$ operator to all such sets, generating a new granularity G'. More formally, $G' = \textit{Select-by-intersect}_k^l(G_1, G_2)$ is the granularity with the label set

$$\mathcal{L}_{G'} = \cup_{i \in \mathcal{L}_{G_2}} \Delta_k^l(\{j \in \mathcal{L}_{G_1} \mid G_1(j) \cap G_2(i) \neq \emptyset\}),$$

and for each $i \in \mathcal{L}_{G'}$,

$$G'(i) = G_1(i).$$

For example, given granularities `week` and `month`, the granularity consisting of the first week of each month (among all the weeks intersecting the month) can be generated by

$$\texttt{FirstWeekOfMonth} = \textit{Select-by-intersect}_1^1(\texttt{week}, \texttt{month}).$$

Again, G' is a label-aligned subgranularity of G_1.

The set operations The set operations are based on the viewpoint that each granularity is a set of granules. Considering each granularity as a set with granules being the elements, we may apply the set operations (i.e., union, intersection, and difference) to existing granularities and generate new granularities. However, not all granularities can be used in set operations. For example, if we union the granularity `week` and `month`, we will not get a valid resulting granularity whatever label sets we use for them. This is because granules of two granularities are not "compatible" in the sense that granules may overlap.

In order to have the set operations as a part of the calendar algebra and to make certain computations easier, we restrict the operand granularities participating in the set operations so that the result of the operation is always a valid granularity: The set operations 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. In the following, we describe the union, intersection, and difference operations of G_1 and G_2, assuming that they satisfy the requirement.

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 $i \in \mathcal{L}_{G'}$,

$$G'(i) = \begin{cases} G_1(i), \, i \in \mathcal{L}_1, \\ G_2(i), \, i \in \mathcal{L}_2 - \mathcal{L}_1. \end{cases}$$

For example, given granularities `Sunday` and `Saturday`, the granularity of the weekend days can be generated by

$$\texttt{WeekendDay} = \texttt{Sunday} \cup \texttt{Saturday}.$$

Note that G_1 and G_2 are label-aligned subgranularities of G'. In addition, if G_1 and G_2 are label-aligned subgranularity of H, then G' is also a label-aligned subgranularity of H. This can be seen from the transitivity of the label-aligned subgranularity relationship (proof is left to the reader).

Intersection. The intersection operation $G_1 \cap G_2$ generates a new granularity G' by taking the common granules from both G_1 and G_2. More formally, $G' = G_1 \cap G_2$ is the granularity with the label set $\mathcal{L}_{G'} = \mathcal{L}_{G_1} \cap \mathcal{L}_{G_2}$, and for each $i \in \mathcal{L}_{G'}$,

$$G'(i) = G_1(i) \text{ (or equivalently } G_2(i)).$$

For example, given the granularity `day1`, which includes exactly the first day of every month, and the granularity `Monday`, the granularity for the days that are both the first day of a month and a Monday can be generated by

$$\texttt{Monday\&day1} = \texttt{Monday} \cap \texttt{day1}.$$

Note that G' is a label-aligned subgranularity of both G_1 and G_2.

Difference. The difference operation $G_1 \setminus G_2$ generates a new granularity G' by excluding the granules of G_2 from those of G_1. More formally, $G' = G_1 \setminus G_2$ is the granularity with the label set $\mathcal{L}_{G'} = \mathcal{L}_{G_1} \setminus \mathcal{L}_{G_2}$, and for each $i \in \mathcal{L}_{G'}$,

$$G'(i) = G_1(i).$$

For example, business days are all the week days that are not federal holidays. Given granularities `Weekday` and `FederalHoliday`,

$$\texttt{BusinessDay} = \texttt{Weekday} \setminus \texttt{FederalHoliday}.$$

Note that G' is a label-aligned subgranularity of G_1.

4 From calendar algebra to periodical set

4.1 The Conversion Process

Our final goal is to provide a correct and effective way to convert calendar expressions into periodical representations. Under appropriate limitations, for each calendar algebra operation, if the periodical descriptions of the operand granularities are known, it is possible to compute the periodical characterization of the resulting granularity.

This result allow us to calculate for any calendar the periodical description of each granularity in terms of the bottom granularity. In fact, by definition, the bottom granularity is fully characterized; hence it is possible to compute the periodical representation of all the granularities that are obtained from operations applied to the bottom granularity. Recursively, the periodical description of all the granularities can be obtained.

The calendar algebra presented in the previous chapter can represent all the granularities that are periodical with finite exceptions (i.e., any granularity G such that bottom groups periodically with finite exceptions into G). Since with the periodical representations defined in Chapter 2 it is not possible to express the finite exceptions, we need to restrict the calendar algebra so that it cannot represent them. This implies to allow the *Subset* operation to be only used as the last step of deriving a granularity. Note that in calendar algebra presented in [NWJ02] there was an extension to the altering tick operation to allow the usage of ∞ as the m parameter (i.e., $G' = Alter^{\infty}_{l,k}(G_2, G_1)$); the resulting granularity has a single exception hence is not periodic. This extension is disallowed here in order to generate periodical granularities only (without finite exceptions).

The conversion process can be divided into three steps: in the first one the period and period label distance are computed; in the second we derive the set $\mathcal{L}^P$ of labels in one period, and in the last one the composition of the explicit granules is computed. For each operation we identify the correct formulas and algorithms for the three steps.

The **first step** consists in computing the period and the period label distance of the resulting granularity. Those values are calculated as a function of the parameters (e.g. the "grouping factor" m, in the *Group* operation) and the operand granularities (actually their periods and period label distances). As stated in Chapter 2, period and period label distance are not unique. Actually an infinite number of them can be identified for a given periodical granularity: in fact if P_G is the period of G and N_G is the associated period label distance, then $\forall \alpha \in \mathbb{Z}^+$, $\alpha \cdot P_G$ is a period of G and $\alpha \cdot N_G$ is the associated period label distance. For this reason we define the notions of *equivalence* and of *minimal period.*

Definition 6. *Two granularities G and H are* equivalent *($G \equiv H$) if $\mathcal{L}_G = \mathcal{L}_H$ and $\forall i \in \mathcal{L}_G$ $G(i) = H(i)$.*

Definition 7. *A granularity G has a* minimal period *if it does not exist a granularity H equivalent to G with period smaller than the period of G.*

The computation of minimal periods is an important aim of our work: the performances of the applications that use periodical representations (as GSTP) strongly depend on the period value of the involved granularities. We conjecture that the formulas presented in this chapter lead to minimal periods.

The **second step** in the conversion process is the identification of the label set of the resulting granularity. In Section 2.4 we pointed out that in order to fully characterize a granularity it is sufficient to identify the labels in any period of the granularity.

In spite of this theoretical result, to perform the computations required by each operation we need the explicit granules of the operand granularities to be "aligned". There are two possible approaches: the first one consist in computing the explicit granules in any period and then recalculate the needed granules in the correct position in order to eventually align them. The second one consists in aligning all the periods containing the explicit granules with a fixed granule in the bottom granularity. After considering both possibilities, for performance reasons, we decided to adopt the second approach. We decided to use $\bot(1)$ as the "alignment point" for all the granularities. A formal definition of the used formalism follows.

Let G be a granularity and i be the smallest positive integer such that $\lceil i \rceil^G$ is defined. We call $l = \lceil i \rceil^G$ and $\overline{\mathcal{L}}_G$ the set of labels of G contained in $l \dots l + N_G - 1$.

Note that this definition of $\overline{\mathcal{L}}_G$ is an instance of the definition of $\mathcal{L}^P$ given in Section 2.4. The definition of $\overline{\mathcal{L}}_G$ provided here is useful for representing G and actually the final goal of this step is to compute $\overline{\mathcal{L}}_G$; however $\overline{\mathcal{L}}_G$ is not suitable for performing the computations. The problem is that if $G(l)$ starts before $\bot(1)$ (i.e., $min(\lfloor l \rfloor^G) < 1$) then the granule $G(l+N_G)$ begins at P_G or before P_G, and hence $G(l+N_G)$ is necessary for the computations; however $l+N_G \notin \overline{\mathcal{L}}_G$.

To solve the problem we introduce the symbol $\hat{\mathcal{L}}_G$ to represent the set of all labels of granules of G that cover one in $\bot(1)\ldots\bot(P_G)$. It is easily seen that if $G(l)$ does not cover $\bot(0)$, then $\hat{\mathcal{L}}_G = \overline{\mathcal{L}}_G$, otherwise $\hat{\mathcal{L}}_G = \overline{\mathcal{L}}_G \cup \{l+N_G\}$. Therefore the conversion between $\overline{\mathcal{L}}$ and $\hat{\mathcal{L}}$ and vice versa is immediate.

The notion of $\hat{\mathcal{L}}$ is still not enough to perform the computations. The problem is that when a granularity G is used as an operand in an operation, the period of the resulting granularity G' is generally bigger than the period of G. Therefore it is necessary to extend the notion of $\hat{\mathcal{L}}_G$ to the period $P_{G'}$ of G' using $P_{G'}$ is spite of P_G in the definition of $\hat{\mathcal{L}}$. The symbol used for this notion is $\hat{\mathcal{L}}_G^{P_{G'}}$.

The idea is that when G is used as the operand in an operation that generates G', $\hat{\mathcal{L}}_G^{P_{G'}}$ is computed from $\overline{\mathcal{L}}_G$. This set is then used by the formula that we provide below to compute $\overline{\mathcal{L}}_{G'}$.

The computation of $\overline{\mathcal{L}}_{G'}$ is performed as follows: if G' is defined by an operation that returns a full integer labeled granularity, then it is sufficient to compute the value of l. Indeed it is easily seen that $\overline{\mathcal{L}}_{G'} = \{i \in \mathbb{Z} | l \leq i \leq l + N_{G'} - 1\}$. If G' is defined by any other algebraic operation, we provide the formulas to compute $\hat{\mathcal{L}}_{G'}$; from $\hat{\mathcal{L}}_{G'}$ we easily derive $\overline{\mathcal{L}}_{G'}$.

Example 3. Figure 3 shows granularities $\bot$, G and H; it is clear that $P_G = P_H = 4$ and $N_G = N_H = 3$. Moreover, $l_G = l_H = 6$ and therefore $\overline{\mathcal{L}}_G = \overline{\mathcal{L}}_H = \{6,7\}$. Since $0 \notin \lfloor 6 \rfloor^G$ then $\hat{\mathcal{L}}_G = \overline{\mathcal{L}}_G$. On the other hand, since $0 \in \lfloor 6 \rfloor^H$, then $\hat{\mathcal{L}}_H = \overline{\mathcal{L}}_H \cup \{6+3\}$.

Suppose that a granularity G' has period $P_{G'} = 8$; then $\hat{\mathcal{L}}_G^{P'_G} = \{6,7,9,10\}$ and $\hat{\mathcal{L}}_H^{P_{G'}} = \{6,7,9,10,12\}$.

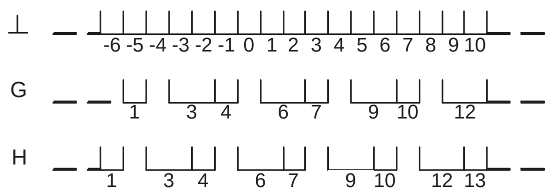

Fig. 3. $\overline{\mathcal{L}}$, l, $\hat{\mathcal{L}}$ and $\hat{\mathcal{L}}^{P_{G'}}$ examples

The **last step** in the conversion process is the computation of the composition of the explicit granules. Once $\overline{\mathcal{L}}_{G'}$ has been computed, it is sufficient to apply, for each label of $\overline{\mathcal{L}}_{G'}$ the formulas presented in Chapter 3.

Computability issues In some of the formulas presented below it is necessary to compute the set S of labels of a granularity G such that $\forall i \in S\ G(i) \subseteq H(j)$ where H is a granularity and j is a specific label of H. Since $\mathcal{L}_G$ contains infinite labels, it is not possible to check, $\forall i \in \mathcal{L}_G$ if $G(i) \subseteq H(j)$. However it is easily seen that $\forall i \in S\ \exists k$ s.t. $G(\lceil k \rceil^G) \subseteq H(j)$. Therefore $\forall i \in S\ \exists k$ s.t. $G(\lceil k \rceil^G)$ is defined and $k \in \lfloor j \rfloor^H$.

Therefore we compute the set S by considering all the labels i of $\mathcal{L}_G$ s.t. $\exists n \in \lfloor j \rfloor^H$ s.t. $\lceil n \rceil^G = i$ and $G(i) \subseteq H(j)$. Since the set $\lfloor j \rfloor^H$ is finite[2], the computation can be performed in a finite time.

The consideration is analogous if S is the set such that $\forall i \in S\ G(i) \supseteq H(j)$ or $\forall i \in S\ (G(i) \cap H(j) \neq \emptyset)$.

4.2 The group operation

Proposition 1 *If $G' = Group_m(G)$, then:*

1. $P_{G'} = \frac{P_G \cdot m}{GCD(m, N_G)}$ *and* $N_{G'} = \frac{N_G}{GCD(m, N_G)}$*;*
2. $l_{G'} = \left(\left\lfloor \frac{l_G - 1}{m} \right\rfloor + 1\right)$*;*
3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = \bigcup_{j=(i-1)\cdot m+1}^{i \cdot m} G(j)$*.*

Example 4. Figure 4 shows an example of the grouping operation: $G' = Group_3(G)$. Since $P_G = 1$ and $N_G = 1$, then $P_{G'} = 3$ and $N_G = 1$. Moreover, since $\overline{\mathcal{L}}_G = \{-7\}$, then $l_G = -7$ and therefore $l_{G'} = -2$ and $\overline{\mathcal{L}}_{G'} = \{-2\}$. Finally $G'(-2) = G(-8) \cup G(-7) \cup G(-6)$ i.e., $G'(-2) = \bot(0) \cup \bot(1) \cup \bot(2)$.

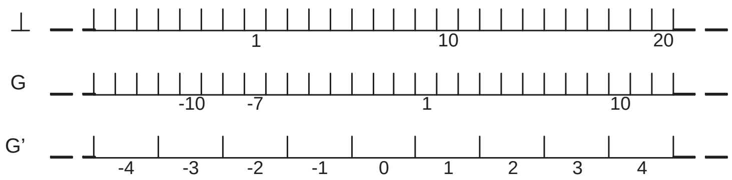

Fig. 4. *Group* operation example

4.3 The altering-tick operation

Proposition 2 *If $G' = Alter^m_{l,k}(G_2, G_1)$ then:*

1.

$$N_{G'} = lcm\left(N_{G_1}, m, \frac{P_{G_2} \cdot N_{G_1}}{GCD(P_{G_2} \cdot N_{G_1}, P_{G_1})}, \frac{N_{G_2} \cdot m}{GCD(N_{G_2} \cdot m, |k|)}\right)$$

[2] With the calendar algebra it is not possible to define granularities having granules that maps to an infinite set of time instants.

and

$$P_{G'} = \left(\frac{N_{G'} \cdot P_{G_1} \cdot N_{G_2}}{N_{G_1} \cdot P_{G_2}} + \frac{N_{G'} \cdot k}{m}\right) \cdot \frac{P_{G_2}}{N_{G_2}}$$

2. $l_{G'} = \lceil l_{G_2} \rceil_{G_2}^{G'}$;
3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = \bigcup_{j=b'_i}^{t'_i} G(j)$ *where* b'_i *and* t'_i *are defined in Section 3.1.*

Referring to step 2., note that when computing $l_{G'}$ the explicit characterization of the granules of G' is still unknown. To perform the operation $\lceil l_{G_2} \rceil_{G_2}^{G'}$ we need to know at least the explicit granules of one of its periods. We choose to compute the granules labeled by $1 \ldots N_{G'}$. When $l_{G'}$ is derived, the granules labeled by $l_{G'} \ldots l_{G'} + N_{G'} - 1$ will be computed so that the explicit granules are aligned to $\perp(1)$ as required.

Example 5. Figure 5 shows an example of the altering tick operation: $G' = Alter^3_{2,1}(G_2, G_1)$. Since $P_{G_1} = 4$, $N_{G_1} = 1$, $P_{G_2} = 4$ and $N_{G_2} = 2$, then $P_{G'} = 6$ and $N_{G'} = 28$. Moreover, since $\overline{\mathcal{L}}_{G_2} = \{-10, -9\}$, then $l_{G_2} = -10$ and therefore $l_{G'} = \lceil -10 \rceil_{G_2}^{G'} = -4$ and hence $\overline{\mathcal{L}}_{G_2} = \{-4, -3, \ldots, 0, 1\}$. Finally $G'(-4) = G_1(-11) \cup G_1(-10) \cup G_1(-9) = \perp(-1) \cup \perp(0) \cup \perp(1) \cup \perp(3) \cup \perp(4)$; analogously we derive $G'(-3)$, $G'(-2)$, $G'(-1)$, $G'(0)$ and $G'(1)$.

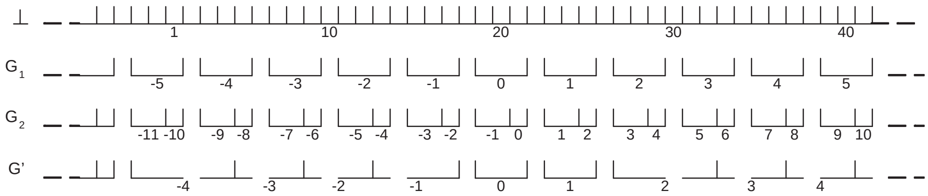

Fig. 5. *Alter* operation example

4.4 The shift operation

Proposition 3 *If* $G' = Shift_m(G)$, *then:*

1. $P_{G'} = P_{G_1}$ *and* $N_{G'} = N_{G_1}$;
2. $l_{G'} = l_G + m$;
3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = G(i - m)$.

Example 6. The shifting operation can easily model time differences. Suppose granularity `USEast-Hour` stands for the hours of US Eastern Time. Since the hours of the US Pacific Time are 3 hours later that those of US Eastern Time, the hours of US Pacific Time can be generated by

$$\texttt{USPacific-Hour} = Shift_{-3}(\texttt{USEast-Hour}).$$

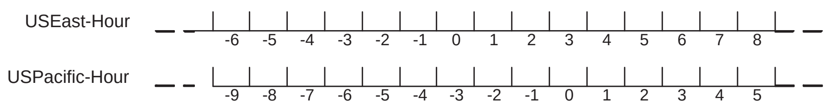

Fig. 6. *Shift* operation example

4.5 The combining operation

Proposition 4 *Given $G' = Combining(G_1, G_2)$, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2})N_{G_1}}{P_{G_1}}$;
2. $\forall i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ *let be* $\widetilde{s}(i) = \{j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} | \emptyset \neq G_2(j) \subseteq G_1(i)\}$; *then* $\hat{\mathcal{L}}_{G'} = \{i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \widetilde{s}(i) \neq \emptyset\}$;
3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = \bigcup_{j \in s(i)} G_2(j)$.

Example 7. Figure 7 shows an example of the operation: $G' = Combine(G_1, G_2)$. Since $P_{G_1} = 6$, $N_{G_1} = 2$, $P_{G_2} = 4$ and $N_{G_2} = 2$, then $P_{G'} = 12$ and $N_{G'} = 4$. Moreover, since $\overline{\mathcal{L}}_{G_1} = \{1\}$ and $0 \in \lfloor 1 \rfloor^{G_1}$, then $\hat{\mathcal{L}}_{G_1} = \{1, 3\}$ and hence $\hat{\mathcal{L}}_{G_1}^{P_{G'}} = \{1, 3, 5\}$. Since $\tilde{s}(i) \neq \emptyset$ for $i \in \{1, 3, 5\}$, then $\hat{\mathcal{L}}_{G'} = \{1, 3, 5\}$; moreover, since $0 \in \lfloor 1 \rfloor^{G'}$, then $\overline{\mathcal{L}}_{G'} = \{1, 3\}$. Finally $s(1) = \{-1, 0\}$ and $s(3) = \{2, 3\}$; consequently, $G'(1) = G_2(-1) \cup G_2(0)$ i.e., $G'(1) = \bot(-1) \cup \bot(0) \cup \bot(1)$ and $G'(3) = G_2(2) \cup G_2(3)$ i.e., $G'(3) = \bot(4) \cup \bot(5) \cup \bot(7)$.

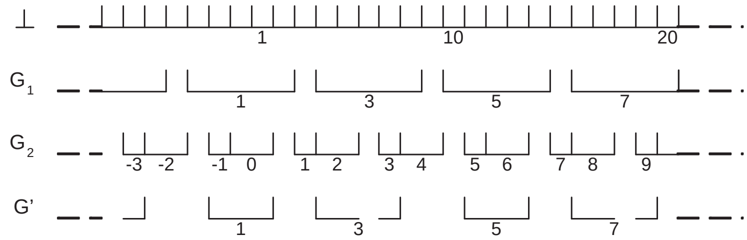

Fig. 7. *Combine* operation example

4.6 The anchored grouping operation

Proposition 5 *Given $G' = Anchored\text{-}group(G_1, G_2)$, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) \cdot N_{G_2}}{P_{G_2}}$;

2.

$$\hat{\mathcal{L}}_{G'} = \begin{cases} \hat{\mathcal{L}}_{G_2}^{P_{G'}}, & \textit{if } l_{G_2} = l_{G_1}, \\ \{l'_{G_2}\} \cup \hat{\mathcal{L}}_{G_2}^{P_{G'}}, & \textit{otherwise,} \end{cases}$$

where l'_{G_2} is the greatest among the labels of $\mathcal{L}_{G_2}$ that are smaller than l_{G_2}.

3. $\forall i \in \overline{\mathcal{L}}_{G'}$ $G'(i) = \bigcup_{j=i}^{i'-1} G_1(j)$ *where i' is the next label of G_2 after i.*

Example 8. Figure 8 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`). Since $P_{\texttt{day}} = 1$ and $P_{\texttt{Sunday}} = 7$, then the period of `USweek` is 7. Moreover since $l_{\texttt{day}} = 11$, $l_{\texttt{Sunday}} = 14$ and $\hat{\mathcal{L}}_{\texttt{Sunday}}^{P_{\texttt{USweek}}} = \{14\}$, then $\hat{\mathcal{L}}_{\texttt{USweek}} = \{7\} \cup \{14\}$. Clearly, since $0 \in \lfloor 7 \rfloor^{\texttt{USweek}}$ then $\overline{\mathcal{L}}_{\texttt{USweek}} = \{7\}$. Finally, $\texttt{USweek}(7) = \bigcup_{j=7}^{13} \texttt{day}(j) = \bigcup_{k=-3}^{3} \bot(k)$.

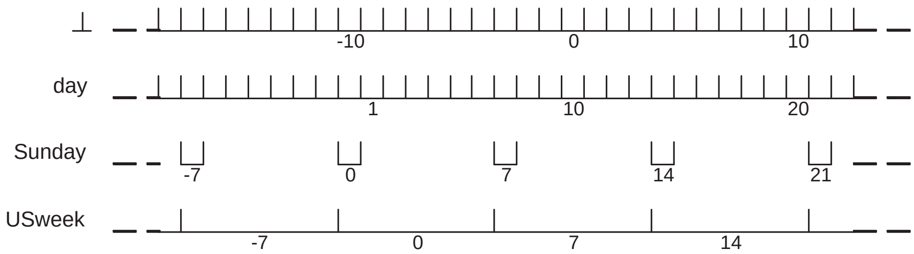

Fig. 8. *Anchored Grouping* operation example

4.7 The subset operation

The *Subset* operation only modifies the operand granularity by introducing the bounds. The period, the period label distance, $\overline{\mathcal{L}}$ and the composition of the explicit granules are not affected.

4.8 The selecting operations

Select-down operation

Proposition 6 *Given $G' = \textit{Select-down}_k^l(G_1, G_2)$, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) \cdot N_{G_1}}{P_{G_1}}$;

2. $\forall i \in \mathcal{L}_{G_2}$ *let*

$$A(i) = \Delta_k^l\left(\{j \in \mathcal{L}_{G_1} | \emptyset \neq G_1(j) \subseteq G_2(i)\}\right).$$

Then

$$\hat{\mathcal{L}}_{G'} = \bigcup_{i \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}} \left\{a \in A(i) | a \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}\right\};$$

3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = G_1(i)$.

Example 9. Figure 9 shows an example of the *Select-down* operation: $G' = \textit{Select-down}_2^1(G_1, G_2)$. Since $P_{G_1} = 4$, $N_{G_1} = 2$ and $P_{G_2} = 6$ then $P_{G'} = 12$ and $N_{G'} = 6$. Moreover, since $\overline{\mathcal{L}}_{G_2} = \{-3\}$ and $0 \in \lfloor -3 \rfloor^{G_2}$, then $\hat{\mathcal{L}}_{G_2} = \{-3, -2\}$ and $\hat{\mathcal{L}}_{G_2}^{P_{G'}} = \{-3, -2, -1\}$. Intuitively, $A(-3) = \{-5\}$, $A(-2) = \{-2\}$ and $A(-1) = \{1\}$. Hence $\hat{\mathcal{L}}_{G'} = \{-5, -2, 1\}$ and therefore, since $0 \in \lfloor -5 \rfloor^{G'}$, $\overline{\mathcal{L}}_{G'} = \{-5, -2\}$. Finally $G'(-5) = G_1(-5) = \perp(0) \cup \perp(1)$ and $G'(-2) = G_1(-2) = \perp(6)$.

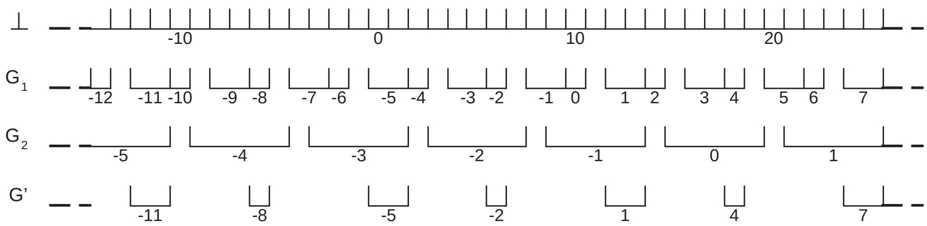

Fig. 9. *Select-down* operation example

Select-up **operation**

Proposition 7 *Given* $G' = \textit{Select-up}(G_1, G_2)$*, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}}$;

2.

$$\hat{\mathcal{L}}_{G'} = \{i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \exists j \in \mathcal{L}_{G_2}\ s.t.\ \emptyset \neq G_2(j) \subseteq G_1(i)\};$$

3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = G_1(i)$.

Example 10. Figure 10 shows an example of the *Select-up* operation: $G' = \textit{Select-up}(G_1, G_2)$. Since $P_{G_1} = 6$, $N_{G_1} = 3$ and $P_{G_2} = 4$ then $P_{G'} = 12$ and $N_{G'} = 6$. Moreover, since $\overline{\mathcal{L}}_{G_1} = \{-3, -2, -1\}$ and $0 \in \lfloor -3 \rfloor^{G_2}$, then $\hat{\mathcal{L}}_{G_1} = \{-3, -2, -1, 0\}$ and $\hat{\mathcal{L}}_{G_1}^{P'_G} = \{-3, -2, -1, 0, 1, 2, 3\}$. Since $G_1(-3) \supseteq G_2(-6)$, $G_1(-1) \supseteq G_2(-4)$ and $G_1(3) \supseteq G_2(0)$ then $\hat{\mathcal{L}}_{G'} = \{-3, -1, 3\}$ and, since $0 \in \lfloor -3 \rfloor^{G'}$, then $\overline{\mathcal{L}}_{G'} = \{-3, 1\}$ Finally $G'(-3) = G_1(-3) = \perp(0) \cup \perp(1)$ and $G'(-1) = G_1(-1) = \perp(4)$.

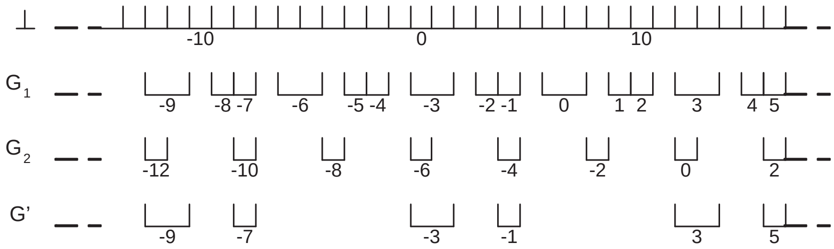

Fig. 10. *Select-up* operation example

Select-by-intersect operation

Proposition 8 *Given* $G' = \textit{Select-by-intersect}^l_k(G_1, G_2)$*, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}}$;
2. *then* $\forall i \in \mathcal{L}_{G_2}$ *let*

$$A(i) = \Delta^l_k \left(\{j \in \mathcal{L}_{G_1} | G_1(j) \cap G_2(i) \neq \emptyset\}\right).$$

then

$$\hat{\mathcal{L}}_{G'} = \bigcup_{i \in \hat{\mathcal{L}}^{P_{G'}}_{G_2}} \left\{a \in A(i) | a \in \hat{\mathcal{L}}^{P_{G'}}_{G_1}\right\}.$$

3. $\forall i \in \overline{\mathcal{L}}_{G'}$ $G'(i) = G_1(i)$.

Example 11. Figure 11 shows an example of the *Select-by-intersect* operation: $G' = \textit{Select-by-intersect}^1_2(G_1, G_2)$. Since $P_{G_1} = 4$, $N_{G_1} = 2$ and $P_{G_2} = 6$ then $P_{G'} = 12$ and $N_{G'} = 6$. Moreover, since $\overline{\mathcal{L}}_{G_2} = \{-3\}$ and $0 \in \lfloor -3 \rfloor^{G_2}$, then $\hat{\mathcal{L}}_{G_2} = \{-3, -2\}$ and $\hat{\mathcal{L}}^{P_{G'}}_{G_2} = \{-3, -2, -1\}$. Intuitively, $A(-3) = \{-6\}$, $A(-2) = \{-2\}$ and $A(-1) = \{0\}$. Hence $\hat{\mathcal{L}}_{G'} = \{-2, 0\}$ and therefore, since $0 \notin \lfloor -5 \rfloor^{G'}$, then $\overline{\mathcal{L}}_{G'} = \{-2, 0\}$. Finally $G'(-2) = G_1(-2) = \perp(6)$ and $G'(0) = G_1(0) = \perp(10)$.

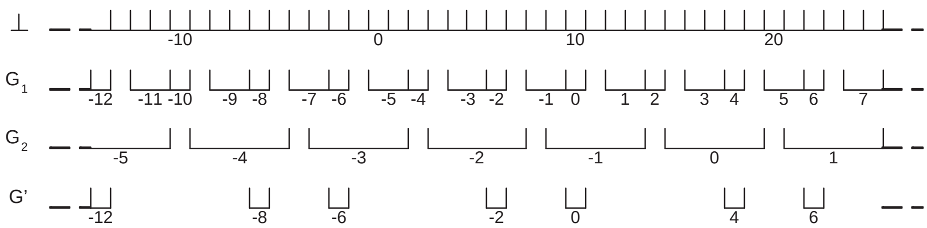

Fig. 11. *Select-by-intersect* operation example

4.9 The set operations

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

Union operation

Proposition 10 *Given $G' = G_1 \cup G_2$, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_2}}{P_{G_2}}$;
2. $\hat{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G_1}^{P_{G'}} \cup \hat{\mathcal{L}}_{G_2}^{P_{G'}}$;
3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = \begin{cases} G_1(i),\ i \in \mathcal{L}_{G_1} \\ G_2(i),\ otherwise, \end{cases}$.

Intersect operation

Proposition 11 *Given $G' = G_1 \cap G_2$, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_2}}{P_{G_2}}$;
2. $\hat{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G_1}^{P_{G'}} \cap \hat{\mathcal{L}}_{G_2}^{P_{G'}}$;
3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = G_1(i)$.

Difference operation

Proposition 12 *Given $G' = G_1 \setminus G_2$, then:*

1. $P_{G'} = lcm(P_{G_1}, P_{G_2})$ *and* $N_{G'} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}} = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_2}}{P_{G_2}}$;
2. $\hat{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G_1}^{P_{G'}} \setminus \hat{\mathcal{L}}_{G_2}^{P_{G'}}$;
3. $\forall i \in \overline{\mathcal{L}}_{G'}\ G'(i) = G_1(i)$.

Example 12. Figure 12 shows an example of the set operations. Note that both G_1 and G_2 are labeled aligned subgranularities of H. Then $G' = G_1 \cup G_2$, $G'' = G_1 \cap G_2$ and $G''' = G_1 \setminus G_2$. Since $P_{G_1} = P_{G_2} = 6$ and $N_{G_1} = N_{G_2} = 6$ then $P_{G'} = P_{G''} = P_{G'''} = 6$ and $N_{G'} = N_{G''} = N_{G'''} = 2$. Moreover, since $\hat{\mathcal{L}}_{G_1} = \{1, 2\}$ and $\hat{\mathcal{L}}_{G_2} = \{2, 3\}$, then $\hat{\mathcal{L}}_{G'} = \{1, 2, 3\}$, $\hat{\mathcal{L}}_{G''} = \{2\}$ and $\hat{\mathcal{L}}_{G'''} = \{1\}$. Finally $G'(1) = G_1(1)$, $G'(2) = G_1(2)$ and $G'(3) = G_2(3)$; $G''(2) = G_1(2)$ and $G'''(1) = G_1(1)$.

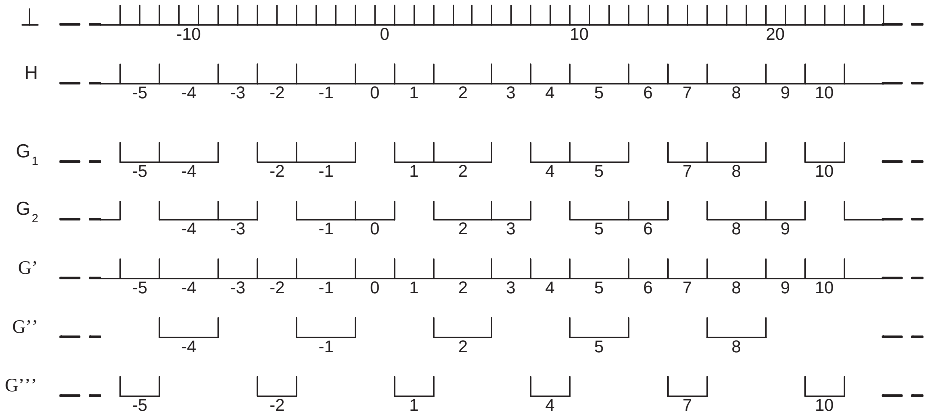

Fig. 12. *Set* operations example

4.10 Relabeling

One of our goals is to use the result of the conversion process to define granularities in the GSTP constraint solver (see 6.1). This system requires the granularities to be labeled by the set of positive integers (i.e., $\mathcal{L} = \mathbb{Z}^+$). Therefore, after the conversion, we have to relabel the resulting granularities. In this section we show how to relabel a granularity G to produce a full integer labeled granularity G'. To produce a granularity G'' such that $\mathcal{L}_{G''} = \mathbb{Z}^+$, we use $G'' = \mathit{Subset}_1^{\infty}(G')$

Note that with the relabeling process some information is lost: for example, if G is a labeled aligned subgranularity of H and $G \not\equiv H$ then after the relabeling G is not a labeled aligned subgranularity of H. The lost information is semantically meaningful in the calendar algebra and therefore the relabeling must be performed only when the granularity will not be used as an operator in an algebraic operation.

Let G be a labeled granularity, i and j integers with $i \in \mathcal{L}_G$ s.t. $G(i) \neq \emptyset$. The relabeling operation $G' = \mathit{Relabel}_i^j(G)$ generates a full-integer labeled granularity G' by relabeling $G(i)$ as $G'(j)$ and relabel the next (and previous) granule of G by the next (previous, respectively) integer. More formally, for each integer k, if $k = j$, then let $G'(k) = G(i)$, and otherwise let $G'(k) = G(i')$ where $G(i')$ is the $|j-k|$-th granule of G after (before, respectively) $G(i)$. If the required $|j-k|$-th granule of G does not exist, then let $G'(k) = \emptyset$. Note the G' is always a full integer labeled granularity.

The relabeling procedure can be implemented in the periodic representation we adopted by computing the value of $l_{G'}$. It is easily seen that once $l_{G'}$ is known the full characterization of G' can be obtained with: $P_{G'} = P_G$; $N_{G'} = R_{G'} = R_G$ and $\overline{\mathcal{L}}_{G'} = \{l_{G'}, l_{G'}+1 \dots l_{G'}+N_{G'}-2, l_{G'}+N_{G'}-1\}$. It is clear that the explicit representation of the granules is not modified.

To compute $l_{G'}$ consider the label $i' = i - \left\lfloor \frac{i-l_G}{N_G} \right\rfloor \cdot N_G$; i' represents the label of $\overline{\mathcal{L}}_G$ such that $i - i'$ is a multiple of N_G. Therefore it is clear that the label

$j' \in \overline{\mathcal{L}}_{G'}$ s.t. $G'(j') = G(i')$ can be computed by $j' = j - \left\lfloor \frac{i - l_G}{N_G} \right\rfloor \cdot N_{G'}$. Finally $l_{G'}$ is obtained with $l_{G'} = j' - |\delta|$ where δ is the distance, in terms of number of granules of G from $G(l_G)$ and $G(i')$.

Example 13. Figure 13 shows an example of the operation: $G' = Relabel^{4}_{33}(G)$. Since $P_G = 4$ and $R_G = 2$ then $P_{G'} = 4$ and $N_{G'} = 2$. Moreover, $i' = 33 - \left\lfloor \frac{33-6}{5} \right\rfloor \cdot 5 = 8$ and $j' = 4 - \left\lfloor \frac{33-6}{5} \right\rfloor \cdot 2 = -6$. Since $l_G = 6$ and $i' = 8$ then $G(i')$ is the next granule of G after $G(l_G)$. Then $\delta = 1$ and hence $l_{G'} = -6 - 1 = -7$. Follows that $\overline{\mathcal{L}}_{G'} = \{-7, -6\}$. Finally $G'(-7) = G(6)$ and $G'(-6) = G(8)$.

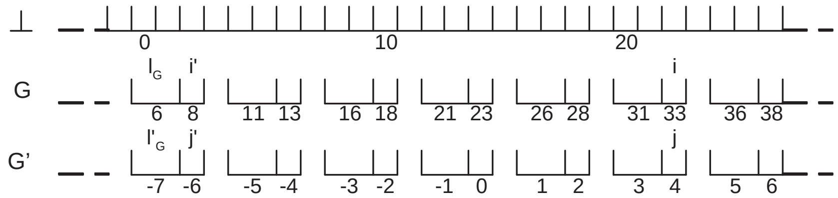

Fig. 13. *Relabeling* example

The GSTP constraint solver impose that the first non empty granule of any granularity ($\perp$ included) is labeled with 1. Therefore when using the relabeling operation for producing granularities for the GSTP, the j parameter must be set to 1. The i have to be equal to the smallest label of G that identify a granule that covers granules of $\perp$ identified by positive labels. By definition of l_G this label is l_G if $min(\lfloor l_G \rfloor^G) > 0$; otherwise is the next label of G after l_G.

4.11 Complexity issues

A detailed complexity analysis is out of the scope of this work. Here we limit to some consideration on time complexity.

For each operation the time necessary to perform the three conversion steps, depends on the operation parameters (e.g. the "grouping factor" m, in the *Group* operation) and on the operand granularities (in particular the period, the period label distance and the number of granules in one period).

A central issue is that if an operand granularity is not the bottom granularity, then its period is a function of the periods of the granularities that are the operands in the operation that defines it. For most of the algebraic operations, in the worst case the period of the resulting granularity is the product of the periods of the operands granularity.

For all operations, the **first step** in the conversion process can be performed in a constant or logarithmic time. Indeed the formulas necessary to derive the period and the period label distance involve (i) standard arithmetic operations, (ii) the computation of the Greatest Common Divisor and (iii) the computation of the least common multiple. Part (i) can be computed in a constant time while (ii) and (iii) can be computed in a logarithmic time using Euclid's algorithm.

For some operations, the **second step** can be performed in constant time (e.g. *Group*, *Shift* or *Anchored-group*) or in linear time (e.g. set operations). For the other operations it is necessary to compute the set S of labels of a granularity G such that $\forall i \in S\ G(i) \subseteq H(j)$ where H is a granularity and $j \in \mathcal{L}_H$ (analogously if S is the set such that $\forall i \in S\ G(i) \supseteq H(j)$ or $\forall i \in S\ (G(i) \cap H(j) \neq \emptyset)$). This computation needs to be performed once for each granule $i \in P_H^{P_{G'}}$. The idea of the algorithm for solving the problem has been presented in 4.1. Several optimizations can be applied to that algorithm but in the worst case (when H covers the entire time domain) it is necessary to perform a number of $\lceil\cdot\rceil^G$ operations linear in the period of the resulting granularity. Since a $\lceil\cdot\rceil^G$ operation requires a time linear in P_G, then the time necessary to perform the second step is quadratic ($O(P_G \cdot P_{G'})$).

The **last step** in the conversion process is performed in linear time with respect to the number of granules in a period of G'. The only exception is the *Alter* operation: also in this case it is necessary to compute the set S of labels of a granularity G such that $\forall i \in S\ G(i) \subseteq H(j)$ where G and H are the first and second operand, respectively. Hence, the time necessary to perform the computation is $O(P_G \cdot P'_G)$.

The complexity analysis of the conversion of a general algebraic expression needs to consider the composition of the operations and hence their complexity.

The last step, relabeling, can be done in linear time.

5 Ensuring period minimality

In this section we show how a minimal periodical representation of a granularity can be generated by any periodical representation of a granularity. We use the notation notion "$\bar{G}^1$", "$\bar{G}^2$", ... (read "representation 1 of G", "representation 2 of G", ...) to refear to different periodical representations of the same granularity. We also indicate with *minimal period* for a granularity G, the value of the period of any of its minimal representations.

First we present an algorithm that, given a granularity representation $\bar{G}^1$, computes the minimal period for G; in Section 5.4 we show how, given this value, a full characterization of a minimal representation can be obtained. The **input** of the algorithm is a periodical representation $\bar{G}^1$; i.e. the period $P_{\bar{G}^1}$, the period label distance $N_{\bar{G}^1}$, and the set of sets S_k with $k = 0 \ldots N_{\bar{G}^1} - 1$ such that, for an arbitrary $\alpha \in \mathbb{Z}$, $S_k = \lfloor k + \alpha \rfloor^G$ if $k \in \mathcal{L}_{\bar{G}^1}$, $S_k = \emptyset$ otherwise.

5.1 The algorithm

The main idea of the algorithm is that if $\bar{G}^1$ is not minimal, then there exists a minimal representation $\bar{G}^2$ such that $P_{\bar{G}^1} = P_{\bar{G}^2} \cdot m$ with $m \in \mathbb{N}^+$. Therefore, the goal of the algorithm is finding m: once it is found the output is simply $\frac{P_{\bar{G}^1}}{m}$. If $\bar{G}^1$ is minimal, then $m = 1$.

Algorithm 1 minimizePeriod

- **Input**: a periodical representation $\bar{G}^1$;
- **Output**: the minimal period for G;
- **Method**:

compute the set S that includes $gcd(P_{\bar{G}^1}, N_{\bar{G}^1}, R_{\bar{G}^1})$ and its factors.
$b := min(\lfloor min(\overline{\mathcal{L}}_{\bar{G}^1})\rfloor^G)$; $t := max(\lfloor max(\overline{\mathcal{L}}_{\bar{G}^1})\rfloor^G)$
for all $n \in S$, from the biggest down to the smallest **do**
    $P := P_{\bar{G}^1}/n$; $N := N_{\bar{G}^1}/n$; failed := **false**
    for ($k = b$; $k < t \wedge$ failed=**false**; k++) **do**
        if ($\lceil k\rceil^G$ is undefined) **then**
            if ($\lceil k+P\rceil^G$ is defined) **then** failed := **true**
        else
            if ($\lceil k+P\rceil^G$ is undefined) **then** failed := **true**
            else if ($\lceil k+P\rceil^G \neq \lceil k\rceil^G + N$) **then** failed := **true**
        end if
    end for
    if (failed= **false**) **then** return P **end if**
end for
return $P_{\bar{G}^1}$

Clearly, m must be a divisor of $P_{\bar{G}^1}$ and we will prove that m must also be a divisor of $N_{\bar{G}^1}$ and $R_{\bar{G}^1}$. Hence, the algorithm first computes the set S of possible values of m, then, for each $n \in S$, it checks if there exists a periodical representation $\bar{G}^3$ such that $P_{\bar{G}^3} = \frac{P_{\bar{G}^1}}{n}$ and $N_{\bar{G}^3} = \frac{N_{\bar{G}^1}}{n}$. Since the value of $P_{\bar{G}^3}$ is inversely proportional to the value of n, the algorithm starts considering the integers in S from the biggest down to the smallest. The execution terminates when the first representation $\bar{G}^3$ is found.

A non trivial part of the algorithm is checking if G can be represented using a period P and a period label distance N. In general, this requires to prove that P and N satisfy the three conditions of Definition 5; however in this particular instance of the problem, it is known that G admits the periodical representation $\bar{G}^1$ and that $\exists k \in \mathbb{N}^+$ s.t. $P = \frac{P_{\bar{G}^1}}{k}$ and $N = \frac{N_{\bar{G}^1}}{k}$. Therefore, it can be derived that the third condition of Definition 5 is always satisfied and the other two conditions are verified if and only if

$$\forall k \in K \ , \ \lceil k+P\rceil^G = \begin{cases} \text{undefined if } \lceil k\rceil^G \text{is undefined} \\ \lceil k\rceil^G + N \text{ otherwise} \end{cases} \qquad (1)$$

where $K = \{j \in \mathcal{L}_\perp | min(\lfloor min(\overline{\mathcal{L}}_{\bar{G}^1})\rfloor^G) \leq j < max(\lfloor max(\overline{\mathcal{L}}_{\bar{G}^1})\rfloor^G)\}$. Note that, since $\mathcal{L}_\perp = \mathbb{Z}$, K is an integers interval and, by definition of K and $\overline{\mathcal{L}}$, $|K| \leq P$. Therefore, the problem can be solved by verifying (1) with k ranging on a finite set.

Example 14. Figure 14 shows granularity G and its non-minimal periodical representation $\bar{G}^1$ (the dotted curly bracket indicates the explicit granules of $\bar{G}^1$). Since $P_{\bar{G}^1} = 12$ and $N_{\bar{G}^1} = R_{\bar{G}^1} = 6$, the algorithm derives $S = \{2, 3, 6\}$. It is not possible to represent G with a period $P = 12/6 = 2$ and a period label distance $N = 6/6 = 1$ since $\lceil 2 \rceil^G$ is defined while $\lceil 2 + P \rceil^G = \lceil 4 \rceil^G$ is undefined. Analogously, it is not possible to represent G with a period $P = 12/3 = 4$ and a period label distance $N = 6/3 = 2$ since $\lceil 2 \rceil^G$ is defined while $\lceil 2 + P \rceil^G = \lceil 2 + 4 \rceil^G$ is undefined. However it is possible to represent G with a period $P = 12/2 = 6$ and a period label distance $N = 6/2 = 3$; For instance, $\lceil 1 \rceil^G = 1$ and $\lceil 1 + P \rceil^G = \lceil 7 \rceil^G = 4 = 1 + N$; $\lceil 4 \rceil^G$ is undefined and $\lceil 4 + P \rceil^G = \lceil 10 \rceil$ is undefined.

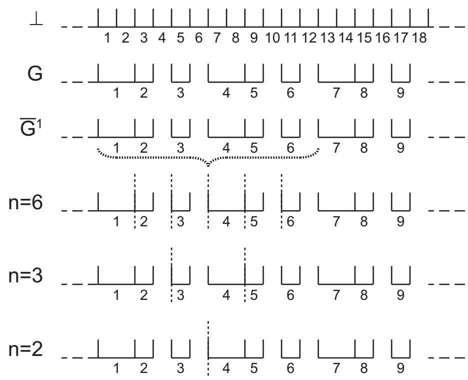

Fig. 14. Graphical representation of the granules involved in Example 14

5.2 Correctness

Theorem 1. *Given a periodical representation of granularity G, the algorithm minimizePeriod computes the minimal period for G.*

Proof. To prove the theorem it is necessary to show three intermediate results.

Lemma 1. *Given a granularity G, $\exists \lambda, \lambda' \in \mathbb{R}^+$ s.t. for each representation $\bar{G}^i$ of G, $\frac{P_{\bar{G}^i}}{N_{\bar{G}^i}} = \lambda$ and $\frac{P_{\bar{G}^i}}{R_{\bar{G}^i}} = \lambda'$.*
Proof in Appendix.

Lemma 2. *If $\bar{G}^1$ and $\bar{G}^2$ are two periodic representations of granularity G and $\bar{G}^1$ is minimal, then $\exists \alpha \in \mathbb{N}^+$ s.t. (i)$P_{\bar{G}^2} = \alpha P_{\bar{G}^1}$, (ii)$N_{\bar{G}^2} = \alpha N_{\bar{G}^1}$; (iii)$R_{\bar{G}^2} = \alpha R_{\bar{G}^1}$.*
Proof in Appendix.

Lemma 3. *Let G be a granularity, $\bar{G}^1$ one of its possible representations and n a positive integer. It is possible to represent G with a period $P = \frac{P_{\bar{G}^1}}{n}$ and a period label distance $N = \frac{N_{\bar{G}^1}}{n}$ if and only if Condition (1) in Section 5.1 is verified.*
Proof in Appendix.

Assuming $\bar{G}^1$ is the representation of G given as input to the algorithm, from Lemma 2 follows that if there exists a minimal representation $\bar{G}^2$ s.t. $P_{\bar{G}^2} < P_{\bar{G}^1}$, then $\exists n \in \mathbb{N}^+$ s.t. $n > 1$, $P_{\bar{G}^1} = nP_{\bar{G}^2}$, $N_{\bar{G}^1} = nN_{\bar{G}^2}$ and $R_{\bar{G}^1} = nR_{\bar{G}^2}$. Clearly the set of possible values for n is the set S containing $gcd(P_{\bar{G}^1}, N_{\bar{G}^1}, R_{\bar{G}^1})$ and its factors.

The condition $P = P_{\bar{G}^1}/n$ with $n \in S$ is necessary but not sufficient for the existence of a periodical representation that has P as the period. We now show that the algorithm correctly verifies if there is a periodical representation having $P_{\bar{G}^1}/n$ as the period. For each $n \in S$, the value of the variable *failed* is first set to **false**; then Condition (1) is checked for each $k \in K$, and, if the condition is not satisfied, then the value of *failed* is set to **true**. Therefore if *failed* = **false** when the **for** cycle terminates, then Condition (1) is verified and, by Lemma 3, there exists a periodical representation having $P_{\bar{G}^1}/n$ as period.

Finally we show that it is correct to stop the evaluation of values in S as soon as a valid representation is found. Indeed, let $S' \subseteq S$ be the set s.t. for each $i \in S'$ there exists a representation of G having $P_{\bar{G}^1}/i$ as period. Then the representation having $P = P_{\bar{G}^1}/max(S')$ as period is minimal. Suppose by contradiction that P is not the minimal representation, then $\exists P'$ s.t. $P' < P$ and P' is the period of the minimal representation of G. From Lemma 2 follows that $\exists m, n \in \mathbb{N}^+$ s.t. $P' = \frac{P}{m}$; and $P = \frac{P_{\bar{G}^1}}{n}$; Then $P' = \frac{P_{\bar{G}^1}}{m \cdot n}$. This leads to a contradiction since $m \cdot n \in S'$, $m \cdot n > n$ and $n = max(S')$.

The last step of the algorithm is correct since if it is not possible to identify a representation for any $n \in S$, then the representation given as input is minimal and its period is returned.

5.3 Time complexity analysis

Before presenting our formal result on the time complexity of the algorithm *minimizePeriod*, we show how it is possible to perform the *up* operation ($\lceil \cdot \rceil$) in constant time. Indeed, from the explicit granules of G, it is possible to create an array A of size $P_{\bar{G}^1}$ that represents how the granules of $\perp$ are mapped into the explicit granules of $\bar{G}^1$. If $b = min(\lfloor min(\overline{\mathcal{L}}_{\bar{G}^1}) \rfloor^G)$, then, for each $j = 0 \ldots P_{\bar{G}^1} - 1$, $A[j] =$ **null** if $\lceil b + j \rceil^G$ is undefined, $A[j] = \lceil b + j \rceil^G$ otherwise[3]. Using this data structure for each i, $\lceil i \rceil^G$ can be computed as $A[j] + \alpha N_{\bar{G}^1}$ where $\alpha = \left\lfloor \frac{i-b}{P_{\bar{G}^1}} \right\rfloor$ and $j = i - b - \alpha P_{\bar{G}^1}$.

Theorem 2. *The worst case time complexity of the algorithm minimizePeriod is $O(n^{\frac{3}{2}})$ where n is the period of the input periodical representation.*

Proof. In the worst case, the number of times the algorithm performs the innermost **for** cycle (Algorithm 1, line 5) is $|S|$. By definition of S, it follows that $|S| < d(P_{\bar{G}^1})$ where $d(n)$ indicates the number of divisors of n. The innermost **for** cycle performs, for each k from b to t, two *up* operations ($\lceil \cdot \rceil$). By definition of b and t, it follows that $t - b \leq P_{\bar{G}^1}$, and since the *up* operation can be executed

[3] Note that A can be built in time $O(P_{\bar{G}^1})$

in constant time, the **for** cycle can be performed in time $O(P_{\bar{G}^1})$. As well known in number theory, if $d(n)$ is the number of divisors of n, then, $d(n) < 2\sqrt{n}$. Hence, the **for** cycle is always executed a number of times less than $2\sqrt{P_{\bar{G}^1}}$, then the thesis follows.

Note that a better upper bound for the dimension of S can be found. Indeed, let $g = gcd(P_{\bar{G}^1}, N_{\bar{G}^1}, R_{\bar{G}^1})$, then $|S| = d(g) - 1$ [4]. Clearly $g \leq P_{\bar{G}^1}$ and, since g is a divisor of $P_{\bar{G}^1}$, $d(g) \leq d(P_{\bar{G}^1})$.

Despite a detailed average-case analysis of time complexity is out of the scope of this paper, note that, in most practical applications of the algorithm, $g << P_{\bar{G}^1}$; Therefore, $d(g) << d(P_{\bar{G}^1})$. Moreover, Theorem 318 in [HW79] states: "$\sum_{i=1}^{n} d(n) \sim n \ln n$"; hence, in the average case, $|S| << d(P_{\bar{G}^1})$ where $d(P_{\bar{G}^1}) \sim \ln(P_{\bar{G}^1})$.

5.4 Characterization of a minimal representation

Here we show how, given a granularity G, its representation $\bar{G}^1$ and its minimal period P, it is possible to fully characterize a minimal representation $\bar{G}^2$ of G.

Clearly $P_{\bar{G}^2} = P$ and, from Lemma 2, $N_{\bar{G}^2} = N_{\bar{G}^1} \cdot P / P_{\bar{G}^1}$. The set of the explicit granules of $\bar{G}^2$ is the set of granules of G having labels in $\overline{\mathcal{L}}_{\bar{G}^2} = \{i \in \overline{\mathcal{L}}_{\bar{G}^1} | min(\overline{\mathcal{L}}_{\bar{G}^1}) \leq i < min(\overline{\mathcal{L}}_{\bar{G}^1}) + N_{\bar{G}^2}\}$. Since $\overline{\mathcal{L}}_{\bar{G}^2} \subset \overline{\mathcal{L}}_{\bar{G}^1}$, the composition of each explicit granule $G(j)$ with $j \in \overline{\mathcal{L}}_{\bar{G}^2}$ in terms of $\perp$ is the same provided in $\bar{G}^1$. Finally, note that if G is bounded, the value of the bounds is the same independently from the representation.

6 Implementation

The theoretical results presented in this work has been exploited to develop a application for converting granularities expressed by calendar expression into equivalent ones expressed by a periodical representation. In order to allow universal access to the software, a web service has been developed. Our final goal is to integrate this application with the existing web service for solving networks of temporal constraints with granularities (GSTP, [BWJ02]) so that users can define granularities appearing in the constraints by calendar algebra expressions. After a sketch presentation of the GSTP System, we describe the calendar converter web service.

6.1 The GSTP System

Figure 15 shows the general architecture of the GSTP system before the developing of the calendar converter web service. There are three main modules: the constraint solver, the GSTP web service, which enables external access to the solver, and a web service client user interface that can be used locally or remotely to design and analyze constraint networks. All data, including time granularity definitions, constraint network specification, algorithm parameters and processing requests are encoded in XML following specific XML schemas.

[4] The "-1" comes from the consideration that the value 1 is not included in S

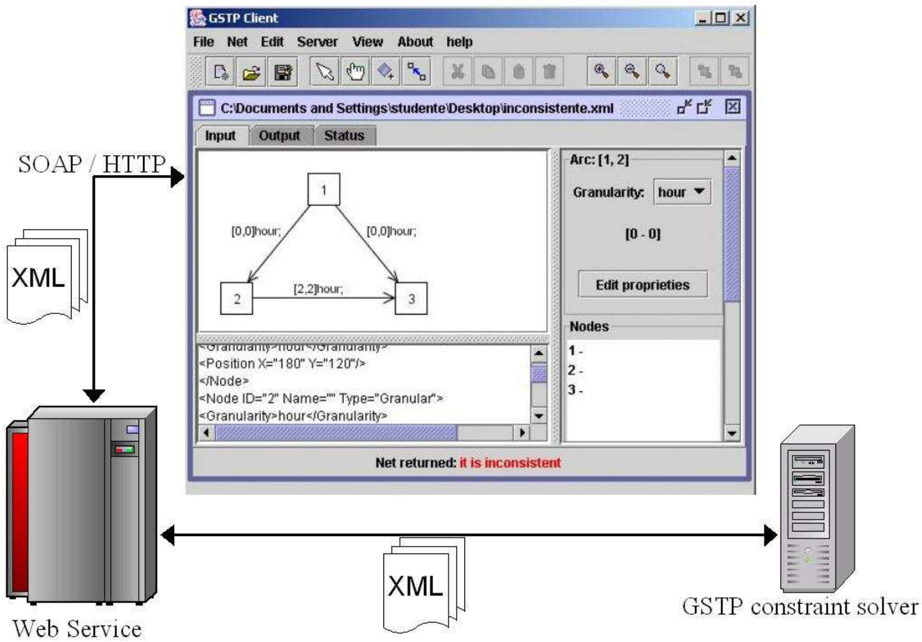

Fig. 15. The GSTP Architecture

The GSTP Constraint Solver The GSTP Constraint Solver is clearly the most complex and innovative component, and the one which required the most implementation efforts. For a formal definition of networks of temporal constraints with granularities and a detailed description of the algorithms and the implementation issues see ([BWJ02] and [BMP04]).

Here we consider only the architectural aspects of the application: it is a performance critic software written in C that runs on an high end server at the university of Milan. It can deal with granularities expressed by a periodical representation only; those granularities are defined "off line" that means that are not part of the input. The consequence is that every time a temporal constraints network is given as input to the application all the granularities appearing in the network must have already been defined.

The only way to add new granularities is to define them in the periodical form. No web service is available for this purpose and therefore it is necessary to work locally on the server. There is a rich set of predefined standard granularities but it is not possible for a user to define his own ones.

GSTP Web Service The Web Service defines, through a WSDL specification, the parameters that can be passed to the constraint solver, including the XML schema for the constraint network specification. The service is exposed to the public web, and despite we provide a specific client application, it can be invoked by different clients or web applications. Therefore, in principle, our service can be easily integrated in any third party software which requires GSTP processing.

The web service application performs three tasks: first of all it validates the parameters by checking if the XML is valid with respect to the XML schema and if the names of the granularities used are already defined. Then it invokes the solver and finally it passes back the results in XML format.

A single service is currently supported even if a number of parameters can be used to specify different versions of the constraint solver algorithms. It is possible, for example, to give up completeness by selecting a variant of the main algorithm in order to have much lower response time, or to use the complete version and possibly set time-out values different from the default ones.

GSTP Client The main goal of the client interface, in addition to remotely interact with the constraint solver through the web service, is to facilitate two tasks: i) the specification and editing of input networks, and ii) the analysis of processed networks.

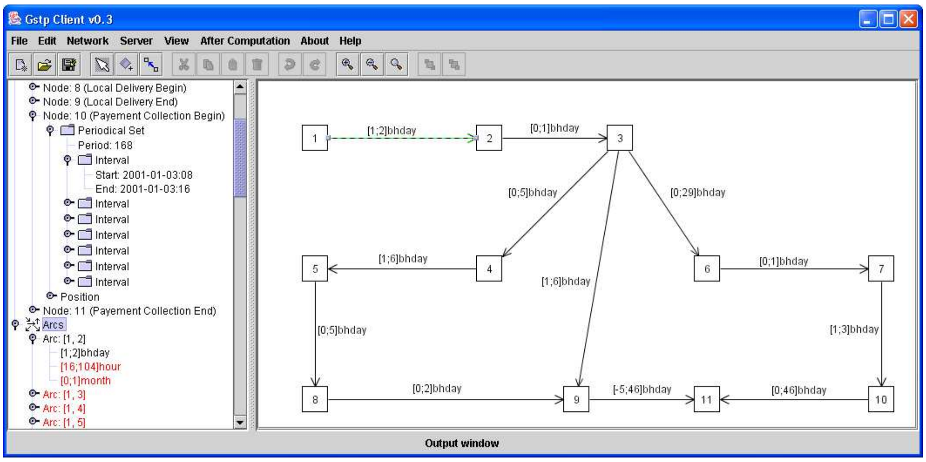

Fig. 16. The view of a processed network in terms of a specific time granularity

For the former task, the GSTP Client supports the user by providing standard functionalities like adding, editing and removing nodes or edges. Networks can also be saved and browsed in XML format.

For the latter, more specialised tools have been developed. In fact the result of the GSTP Constraint Solver is a fully connected network having each arc possibly labeled by one constraint for each of the granularities appearing in the input network. It is clear that is practically infeasible to graphically show all this information in a single screenshot in a way that is still useful to the user.

Therefore some functionalities have been introduced: first of all zooming and scrolling features allow to examine large networks, while nodes can be automatically disposed in order not to overlap with each other or to preserve the position they had in the input network. Moreover it is possible to selectively hide and show information from the network: in particular it is possible to have views of the network in terms of specific single or set of time granularities. Figure 16 shows the GSTP Client interface showing the result of a GSTP constraint solver computation. The nodes are disposed as they were in the input network, and the only edges that are shown are the ones that were explicit in the input network. All the constraints are hidden except those in terms of the granularity `bhday` (the business hour day, i.e., the working hours during the business days).

A specific functionality has been introduced in order to show a network solution if the network is found to be consistent (see Figure 17).

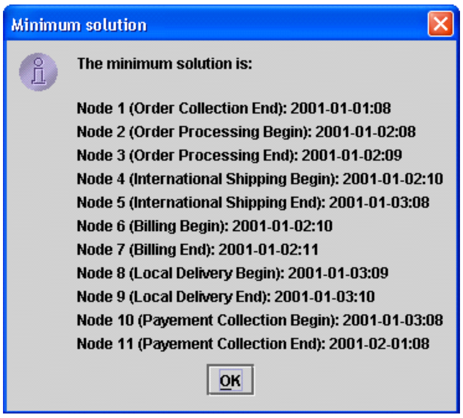

Fig. 17. Displaying a solution

For a more detailed description of the functionalities of the client application see [BMP03].

6.2 Conversion Software Design and Implementation

Architecture The Calendar Converter software has been implemented as a web service. It provides the service *CalendarConversion* that receives a XML description of a calendar expressed by calendar algebra expressions and returns the XML description of the granularities in the periodical representation.

The Calendar Converter web service does not have any functionality for storing the converted calendars and for communicating with the GSTP constraint solver. Those functionalities are necessary for the GSTP system to allow the definition of new calendars and will be performed by the GSTP web service.

Figure 18 shows the architecture of the GSTP system interacting with the Calendar Converter system. Note that a Calendar Converter client has not been developed yet; it functionalities can include a user friendly definition of calendar algebra expressions and a graphical visualization of the corresponding granularities expressed by a periodical representation.

Moreover the GSTP web service still needs to be updated: new services like adding or removing a calendar need to be added; in order to perform those services, the application will have to invoke the Calendar Converter web service.

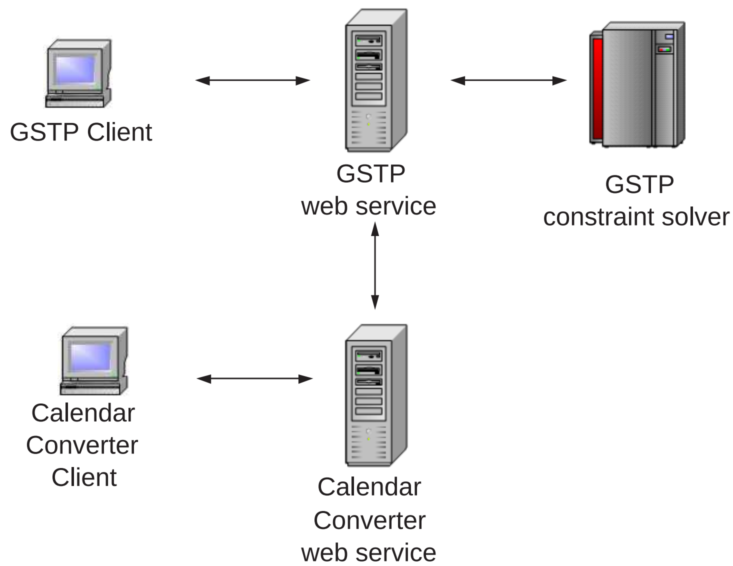

Fig. 18. Integration between GSTP and Calendar Converter systems

XML schemas **The input** of the Calendar Converter web service consists in a Calendar where each granularity is expressed as an algebraic expression. The two XSD schemas used for representing this information has been designed in collaboration with Prof. Wang's research group at the University of Vermont.

A schema (see B.1) defines the algebraic operation used in the calendar algebra; it is a general schema that is valid for every calendar. This schema is imported in the second one where the names of the granularities composing a given calendar are defined; hence several versions of this schema exist, one for each set of different granularity names. See B.2 for an example of this second schema. The XML used to represent a particular calendar (i.e., a set of granularity names and their calendar algebra representation) is an instances of the second schema (see B.3 for an example).

The output of the Calendar Converter web service consists in a Calendar where each granularity is expressed with a periodical representation. The XML schema represents, for each granularity G, the period, the period label distance and, for each label $i \in \overline{\mathcal{L}}_G$ the composition of $G(i)$ in terms of the bottom granularity. If G is a bounded granularity, the bounds are indicated. See Section B.4 for the XSD schema and Section B.5 for an XML example.

Technologies For the implementation of the calendar converter web service we adopted the same technologies used in the GSTP web service. A detailed analysis was performed before the implementation of the GSTP web service with the aim of choosing the best technologies for the application. Since the calendar converter web service requirements are similar to the GSTP web service ones, there are no reason for using different technologies.

The application has been written in *Java* (`www.java.sun.com`). We used *Axis* as the SOAP engine (`ws.apache.org/axis/`) and we deployed the web service in the *Tomcat* application server (`jakarta.apache.org/tomcat/`).

Implementation In this section we present the packages structure and we describe the relationships among the classes.

Packages Structure. The main package *calendarConverter* contains three packages:

- *calAlgCalendar* that contains the classes necessary to represent a calendar having granularities expressed as calendar expressions;
- *perGranCalendar* that contains the classes necessary to represent a calendar having granularities expressed by periodical granules;
- *conversions* that contains, for every calendar algebra operation, a class that performs the conversion for that operation.

The *calendarConverter* contains also three classes: *Converter* that contains the method *convertCalendar* available from the web; *Runner* that is used for locally testing the application and *Shared* that contains some static methods used by other classes.

The package *calAlgCalendar* contains the package *operations* that include the classes necessary to represent every calendar algebra operation.

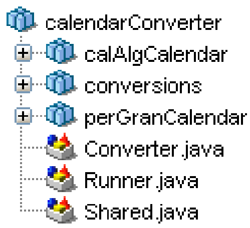

Fig. 19. Packages Structure

Classes Structure. Figure 20 shows the structure of the classes for the calAlgCalendar package. A calendar represented by the calendar algebra is implemented by the *CalAlgCalendar* that contains the name of the bottom granularity and a set of granularities expressed as calendar algebra expressions. Each *CalAlgGranularity* is identified by a name and a reference to an operation. The concept of

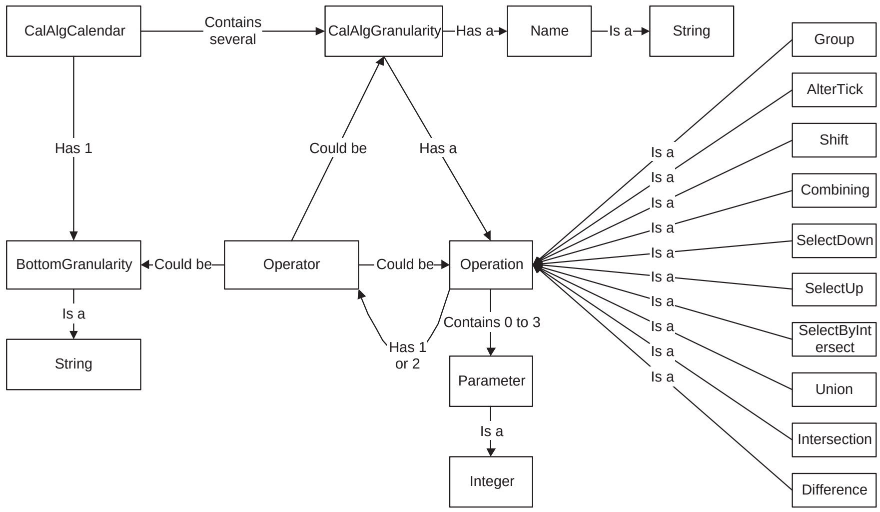

Fig. 20. Classes structure for the calAlgCalendar package

calendar algebra operation is central in this package; for each calendar algebra operation exists a specific class that represents it. However it also exists a generalization class called *Operation* that contains the integer parameters and the references to the operators. A *Operator* can be the bottom granularity or an already defined granularity or an operation.

Figure 21 shows the structure of the classes for the perGranCalendar package. A calendar expressed with the periodical representation is implemented by the *PerGranCalendar* class that contains the bottom granularity and a set of granularities expressed with the periodical representation. Each *PerGranGranularity* is identified by a name and contains the value for the period, the period label distance, the lower and upper bounds and a list of granules. Each *Granule* contains several *Interval* objects each one indicating the starting and ending points of the interval.

Performances The developed software has not been optimized yet. However the time performance we obtained are encouraging: a standard Gregorian calendar that has hour as the bottom granularity and that deals with leap year and leap year exception in a period of 400 years can be converted in few seconds on a standard computer (10 seconds are necessary on a PIII 1Ghz).

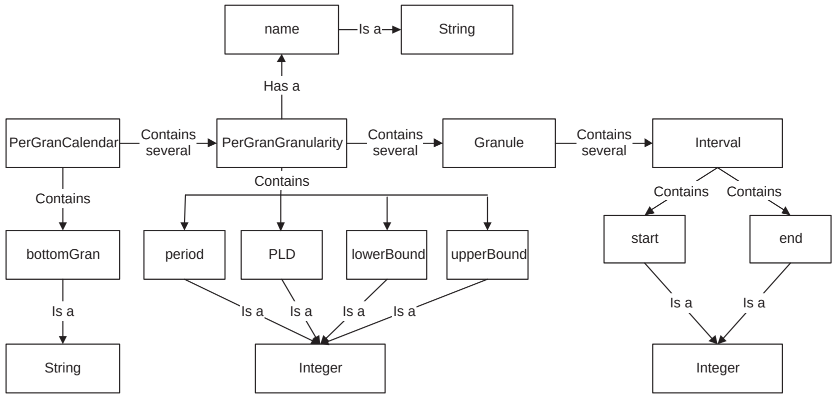

Fig. 21. Classes structure for the perGranCalendar package

Moreover several optimization can be introduced in the code and hence we think that the performance could be significantly improved.

7 Conclusion and future work

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. This result allowed us to implement a web service to convert granularities expressed in terms of calendar algebra expressions into ones expressed in terms of periodical granularities. In practice user friendly formalism can be used to define granularities while their corresponding periodical representation is used for processing. An example of the application of the principle is the definition of granularities constraints networks and their processing by the GSTP system [BWJ02].

As future work we are investigating in more depth the issue of computational complexity discussed in section 4.11. Regarding implementation and system integration we plan to develop a calendar converter client that will help the user to compose the calendar expressions and to invoke the calendar converter web service. Moreover it could graphically show the granularities obtained with the conversion strongly improving the readability of the periodical representation.

References

[BMW04] C. Bettini, S. Mascetti, X. Wang. Mapping Calendar Expressions into Periodical Granularities. In Proc. of *11th International Symposium on Temporal Representation and Reasoning*, IEEE Computer Society, to appear.

[BMP04] C. Bettini, S. Mascetti. V. Pupillo A system prototype for solving multi-granularity temporal CSP. In Workshop Notes of the ERCIM-COLOGNET International Workshop on Constraint satisfaction and Constraint Logic Programming, Lausanne, June, 2004.

[BM04] C. Bettini, S. Mascetti Towards Minimal Periodical Representation of Calendar Expressions In Workshop Notes of the ECAI International Workshop on Spatial and Temporal Reasoning, Valencia, August 2004.
[HW79] G. H. Hardy, E. M. Wright, *An Introduction to the Theory of Numbers.* Oxford University Press, 1960.
[Ter03] Paolo Terenziani, Symbolic User-Defined Periodicity in Temporal Relational Databases. *IEEE Trans. on Knowledge and Data Engineering*, 15(2): 489–509, 2003.
[BMP03] C. Bettini, S. Mascetti, V. Pupillo. GSTP: A Temporal Reasoning System Supporting Multi-Granularity Temporal Constraints, in Proc. of Int. Joint Conference on Artificial Intelligence (IJCAI), (Intelligent Systems Demonstrations), pp. 1633-1634, Morgan Kaufmann, San Francisco, CA, 2003.
[BWJ02] C. Bettini, X. Wang, S. Jajodia, Solving Multi-Granularity constraint networks, *Artificial Intelligence*, 140(1-2):107–152, 2002.
[NWJ02] P. Ning, X. Wang, S. Jajodia. An Algebraic Representation of Calendars. *Annals of Mathematics and Artificial Intelligence* 36(1-2): 5–38, 2002.
[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.
[BWJ98] C. Bettini, X. Wang, S. Jajodia. A General Framework for Time Granularity and its Application to Temporal Reasoning. *Annals of Mathematics and Artificial Intelligence* 22(1,2):29–58, 1998.
[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.
[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.

A Proofs

A.1 Transitivity of the *Periodically Groups Into* relationship

In order to prove the correctness of the conversions of algebraic expressions into periodical sets, it is useful to have a formal result about the transitivity of the *periodically groups into* relation. In addition to transitivity of $\trianglelefteq$, Theorem 1 also say something about period values.

Theorem 1 *Let G and H be two unbounded granularities such that G is periodic in terms of the bottom granularity (i.e., $\bot \trianglelefteq G$) and H is periodic in terms of G (i.e., $G \trianglelefteq H$). Let P_H^G be the period of H in terms of granules of G, and N_G the period label distance of G. Then, if $P_H^G = \alpha N_G$ for some positive integer α, then H is periodic in terms of the bottom granularity (i.e., $\bot \trianglelefteq H$) and $P_H = \alpha P_G$.*

Proof. Since by hypothesis $G \trianglelefteq H$ and $P_H^G = \alpha N_G$, $\forall i$ if $H(i) = \bigcup_{r=0}^{n_i} G(i_r)$, then $\exists \beta \in \mathbf{N}$ such that $H(i+\beta) = \bigcup_{r=0}^{n_i}(i_r + \alpha N_G)$. This can be also written as follows:
if

$$H(i) = G(i_0) \cup ... \cup G(i_{n_i}) \tag{2}$$

then $\exists \beta \in \mathbf{N} s.t.$:

$$H(i+\beta) = G(i_0 + \alpha N_G) \cup ... \cup G(i_{n_i} + \alpha N_G) \quad (3)$$

Since $\bot \trianglelefteq G$, if

$$G(i_j) = \bigcup_{k=0}^{\tau_{i_j}} \bot(i_{j,k}) \quad (4)$$

then

$$G(i_j + N_G) = \bigcup_{k=0}^{\tau_{i_j}} \bot(i_{j,k} + P_G) \quad (5)$$

This can be clearly extended by using αN_G instead of N_G.

$$G(i_j + \alpha N_G) = \bigcup_{k=0}^{\tau_{i_j}} \bot(i_{j,k} + \alpha P_G) \quad (6)$$

Rewriting (2) substituting $G(i_j)$ according to (4) and rewriting (3) substituting $G(i_j + \alpha N_G)$ according to (6), we obtain:
if $H(i) = \underbrace{\bot(i_{0,0}) \cup \ldots \cup \bot(i_{0,\tau_{i_0}})}_{G(i_0)} \cup \ldots \cup \underbrace{\bot(i_{n_i,0}) \cup ... \cup \bot(i_{n_i,\tau_{i_{n_i}}})}_{G(i_{n_i})}$
then $H(i+\beta) = \underbrace{\bot(i_{0,0} + \alpha P_G) \cup \ldots \cup \bot(i_{0,\tau_{i_0}} + \alpha P_G)}_{G(i_0+\alpha N_G)} \cup \ldots$
$\cup \underbrace{\bot(i_{n_i,0} + \alpha P_G) \cup \ldots \cup \bot(i_{n_i,\tau_{i_{n_i}}} + \alpha P_G)}_{G(i_{n_i}+\alpha N_G)}$
Hence the second condition of Definition 5 is satisfied. The third one is always satisfied for unbounded granularities. Since the period label distance of H in terms of $\bot$ denoted by β in the above proof has been shown to be equal to the period label distance of H in terms of G, then by the hypothesis that $G \trianglelefteq H$, for each label i of H, $i + N_H$ is also a label of H. Therefore the first condition of Definition 5 is satisfied too. Hence, $\bot \trianglelefteq H$ with $P_H = \alpha P_G$ and $N_H = \beta$.

A.2 Proof of Proposition 1

Part 1 From the definition of the *Group* operation, for all $i \in \mathbf{N}$:

$$G'(i) = \bigcup_{j=(i-1)m+1}^{im} G(j) = G(im-m+1) \cup \ldots \cup G(im) = G(\lambda) \cup \ldots \cup G(\lambda+m-1)$$

with $\lambda = im - m + 1$. Furthermore, $\forall k \in \mathbf{N}$:

$$G'(i+k) = \bigcup_{j=(i+k-1)m+1}^{(i+k)m} G(j) = G(im + km - m + 1) \cup \ldots \cup G(im + km) =$$

$$= G(\lambda + km) \cup \ldots \cup G(\lambda + km + m - 1)$$

Hence,

$$\text{If } G'(i') = \bigcup_{r=0}^{m-1} G(\lambda + r) \text{ then } G'(i' + k) = \bigcup_{r=0}^{m-1} G(\lambda + r + km). \quad (7)$$

This holds for each k. If we use $k = \frac{N_G}{GCM(m,N_G)}$ (note that $k \in \mathbb{N}$), then all the hypothesis of Theorem 1 are satisfied: (i) $\perp \trianglelefteq G$ (for hypothesis); (ii) $G \trianglelefteq G'$ (since $G \trianglelefteq G'$, $\mathcal{L}_{G'} = \mathbb{Z}$, and (7) holds); (iii) $P_{G'}^{G} = \frac{m \cdot N_G}{GCM(m,N_G)}$ (since we use $k = \frac{N_G}{GCM(m,N_G)}$ and, from (7) we know that $P_{G'}^{G} = km$). Therefore, by Theorem 1, $\perp \trianglelefteq G'$ with $P_{G'} = \frac{mP_G}{GCM(m,N_G)}$ and $N_{G'} = \frac{N_G}{GCM(m \cdot N_G)}$.

Part 2 By definition of l, we need to show that $G'\left(\left\lfloor \frac{l_G-1}{m} \right\rfloor + 1\right) = \bigcup_{j=b}^{t} G(j)$ with $b \leq l_G \leq t$.

From the definition of *Group* operation, $G'(i) = \bigcup_{j=(i-1)\cdot m+1}^{i \cdot m}$; hence:

$$G'\left(\left\lfloor \frac{l_G - 1}{m} \right\rfloor + 1\right) = \bigcup_{j=\left\lfloor \frac{l_G-1}{m} \right\rfloor \cdot m+1}^{\left(\left\lfloor \frac{l_G-1}{m} \right\rfloor + 1\right)\cdot m} G(j)$$

We prove the thesis showing that (1) $\left\lfloor \frac{l_G-1}{m} \right\rfloor \cdot m + 1 \leq l_G$ and that (2) $\left(\left\lfloor \frac{l_G-1}{m} \right\rfloor + 1\right) \cdot m \geq l_G$.

(1) Since $\left\lfloor \frac{l_G-1}{m} \right\rfloor \leq \frac{l_G-1}{m}$, hence $\left\lfloor \frac{l_G-1}{m} \right\rfloor \cdot m + 1 \leq l_G$

(2) First we prove that $\left\lfloor \frac{l_G-1}{m} \right\rfloor \geq \frac{l_G}{m} - 1$. Since $\left\lfloor \frac{l_G-1}{m} \right\rfloor = \frac{l_G-1-[(l_G-1) mod\ m]}{m}$ we have to prove that $\frac{l_G-1-[(l_G-1) mod\ m]}{m} \geq \frac{l_G}{m} - 1$. From this follows that $-[(l_G - 1) mod\ m] \geq -m + 1$; therefore $(l_G - 1) mod\ m \leq m - 1$ that is clearly true. Since $\left\lfloor \frac{l_G-1}{m} \right\rfloor \geq \frac{l_G}{m} - 1$ it is trivial that $\left(\left\lfloor \frac{l_G-1}{m} \right\rfloor + 1\right) \cdot m \geq l_G$.

A.3 Proof of Proposition 2

Part 1 Proof sketch

We show that $G_2 \trianglelefteq G'$ with $P_{G'}^{G_2} = \alpha N_{G_2}$ and then we apply Theorem 1 to obtain the thesis. In particular we use

$$\Delta = lcm\left(N_{G_1}, m, \frac{P_{G_2} \cdot N_{G_1}}{GCD(P_{G_2} \cdot N_{G_1}, P_{G_1})}, \frac{N_{G_2} \cdot m}{GCD(N_{G_2} \cdot m, |k|)}\right)$$

and

$$\alpha = \left(\frac{\Delta \cdot P_{G_1} \cdot N_{G_2}}{N_{G_1} \cdot P_{G_2}} + \frac{\Delta \cdot k}{m}\right) \cdot \frac{P_{G_2}}{N_{G_2}}$$

such that, for each i, if $\exists j, k : G'(i) = \bigcup_{r=0}^{k} G_2(j + r)$, then $G'(i + \Delta) = \bigcup_{r=0}^{k} G_2(j + r + \alpha N_{G_2})$.

Given an arbitrary granule $G'(i)$, we show that $G'(i+\Delta)$ is the union of granules that can be obtained adding αN_{G_2} to the index of each granule of G_2 contained in $G'(i)$. Note that $i+\Delta \in \mathcal{L}_{G'}$ since G' is full-integer labeled. In order to show that this is correct we consider the way granules of G' are constructed by definition of altering-tick. More precisely, we compute the difference between the label $b'_{i+\Delta}$ of the first granule of G_2 included in $G'(i+\Delta)$ and the label b'_i of the first granule of G_2 included in $G'(i)$; we show that this difference is equal to the difference between the label $t'_{i+\Delta}$ of the last granule of G_2 included in $G'(i+\Delta)$ and the label t'_i of the last granule of G_2 included in $G'(i)$. This fact together with the consideration that G_2 is a full integer labeled granularity, leads to the conclusion that $G'(i)$ and $G'(i+\Delta)$ have the same number of granules. It is then clear that the above computed label differences are also equal to the difference between the label of an arbitrary n-th granule of G_2 included in $G'(i+\Delta)$ and the label of the n-th granule of G_2 included in $G'(i)$. If this difference is $b'_{i+\Delta} - b'_i$, then we have: if $\exists j,k : G'(i) = \bigcup_{r=0}^{k} G_2(j+r)$, then $G'(i+\Delta) = \bigcup_{r=0}^{k} G_2\left(j+r+\left(b'_{i+\Delta}-b'_i\right)\right)$. By showing that $b'_{i+\Delta}-b'_i$ is a multiple of N_{G_2} the thesis follows.

Proof details

Assume $G_1(i) = \bigcup_{j=b_i}^{t_i} G_2(j)$ and $G_1(i+\Delta) = \bigcup_{j=b_{i+\Delta}}^{t_{i+\Delta}} G_2(j)$. We need to compute $b'_{i+\Delta} - b'_i$. From the definition of the altering tick operation:

$$b'_i = \begin{cases} b_i + \left(\left\lfloor \frac{i-l}{m} \right\rfloor\right) k & \text{if } i = \left(\left\lfloor \frac{i-l}{m} \right\rfloor\right) m + l, \\ b_i + \left(\left\lfloor \frac{i-l}{m} \right\rfloor + 1\right) k & \text{otherwise.} \end{cases} \tag{8}$$

and

$$b'_{i+\Delta} = \begin{cases} b_{i+\Delta} + \left(\left\lfloor \frac{i+\Delta-l}{m} \right\rfloor\right) k & \text{if} i+\Delta = \left(\left\lfloor \frac{i+\Delta-l}{m} \right\rfloor\right) m + l, \\ b_{i+\Delta} + \left(\left\lfloor \frac{i+\Delta-l}{m} \right\rfloor + 1\right) k & \text{otherwise.} \end{cases} \tag{9}$$

Note that if $i = \left(\left\lfloor \frac{i-l}{m} \right\rfloor\right) m + l$, then $i+\Delta = \left(\left\lfloor \frac{i+\Delta-l}{m} \right\rfloor\right) m + l$. Indeed, $\left(\left\lfloor \frac{i+\Delta-l}{m} \right\rfloor\right) m + l = \left(\left\lfloor \frac{i-l}{m} + \frac{\Delta}{m} \right\rfloor\right) m + l$ and, since Δ is a multiple of m, then $\left(\left\lfloor \frac{i-l}{m} + \frac{\Delta}{m} \right\rfloor\right) m + l = \left(\frac{\Delta}{m} + \left\lfloor \frac{i-l}{m} \right\rfloor\right) m + l = \Delta + \left(\left\lfloor \frac{i-l}{m} \right\rfloor\right) m + l$.

Hence, to compute $b'_{i+\Delta} - b'_i$ we should consider two cases:

$$b'_{i+\Delta} - b'_i = \begin{cases} b_{i+\Delta} + \left(\left\lfloor \frac{i+\Delta-l}{m} \right\rfloor\right) k - b_i - \left(\left\lfloor \frac{i-l}{m} \right\rfloor\right) k & \text{if } i = \left(\left\lfloor \frac{i-l}{m} \right\rfloor\right) m + l \\ b_{i+\Delta} + \left(\left\lfloor \frac{i+\Delta-l}{m} \right\rfloor + 1\right) k - b_i - \left(\left\lfloor \frac{i-l}{m} \right\rfloor + 1\right) k & \text{otherwise.} \end{cases} \tag{10}$$

In both cases (again considering the fact that Δ is a multiple of m):

$$b'_{i+\Delta} - b'_i = (b_{i+\Delta} - b_i) + \frac{\Delta \cdot k}{m} \tag{11}$$

We are left to compute $b_{i+\Delta} - b_i$, i.e., the distance in terms of granules of G_2, between $G_2(b_i)$ and $G_2(b_{i+\Delta})$. Since, by hypothesis, $G_1(i) = \bigcup_{j=b_i}^{t_i} G_2(j)$ and

$G_1(i+\Delta) = \bigcup_{j=b_{i+\Delta}}^{t_{i+\Delta}} G_2(j)$, then the first granule of $\perp$ making $G_2(b_i)$ and the first granule of $\perp$ making $G_1(i)$ are the same granule. The same can be observed for the first granule of $\perp$ making $G_2(b_{i+\Delta})$ and the first granule of $\perp$ making $G_1(i+\Delta)$. More formally:

$$min \lfloor b_i \rfloor^{G_2} = min \lfloor i \rfloor^{G_1}$$

and

$$min \lfloor b_{i+\Delta} \rfloor^{G_2} = min \lfloor i+\Delta \rfloor^{G_1}$$

Hence, we have:

$$min \lfloor b_{i+\Delta} \rfloor^{G_2} - min \lfloor b_i \rfloor^{G_2} = min \lfloor i+\Delta \rfloor^{G_1} - min \lfloor i \rfloor^{G_1} \tag{12}$$

We have shown that the difference between the index of the first granule of $\perp$ making $G_2(b_{i+\Delta})$ and the index of the first granule of $\perp$ making $G_2(b_i)$ is equal to the difference between the index of the first granule of $\perp$ making $G_1(i+\Delta)$ and the index of the first granule of $\perp$ making $G_1(i)$. Then, we need to compute the the difference between the index of the first granule of $\perp$ making $G_1(i+\Delta)$ and the index of the first granule of $\perp$ making $G_1(i)$. Since $\perp \trianglelefteq G_1$, for each i, if $\exists j, \tau : G_1(i) = \bigcup_{r=0}^{\tau} \perp(i+r)$, then $G_1(i+\Delta) = \bigcup_{r=0}^{\tau} \perp(i + \frac{\Delta \cdot P_{G_1}}{N_{G_1}})$. Hence, this difference has value $\frac{\Delta \cdot P_{G_1}}{N_{G_1}}$, and for what shown above this is also the value of the difference between the index of the first granule of $\perp$ making $G_2(b_{i+\Delta})$ and the index of the first granule of $\perp$ making $G_2(b_i)$. Then, since $\perp \trianglelefteq G_2$ with period P_{G_2} and since $\frac{\Delta \cdot P_{G_1}}{N_{G_1}}$ is a multiple of P_{G_2}, we have that, if:

$$\perp(j) \subseteq G_2(i)$$

then:

$$\perp(j + \frac{\Delta \cdot P_{G_1}}{N_{G_1}}) \subseteq G_2(i + \frac{\Delta \cdot P_{G_1} \cdot N_{G_2}}{N_{G_1} \cdot P_{G_2}})$$

Thus, $b_{i+\Delta} - b_i = \frac{\Delta \cdot P_{G_1} \cdot N_{G_2}}{N_{G_1} \cdot P_{G_2}}$.

Reconsidering 11:

$$b'_{i+\Delta} - b'_i = \frac{\Delta \cdot P_{G_1} \cdot N_{G_2}}{N_{G_1} \cdot P_{G_2}} + \frac{\Delta \cdot k}{m}.$$

Analogously we can compute $\frac{\Delta \cdot P_{G_1} \cdot N_{G_2}}{N_{G_1} \cdot P_{G_2}} + \frac{\Delta \cdot k}{m}$.

Thus, $b'_{i+\Delta} - b'_i = t'_{i+\Delta} - t'_i$; hence $t_{i+\Delta} - b_{i+\Delta} = t_i - b_i$. Since G_2 is a full integer labeled granularity, then $G'(i)$ and $G'(i+\Delta)$, are formed by the same number of granules.

Since we now know $G'(i+\Delta) = \bigcup_{j=b'_{i+\Delta}}^{t'_{i+\Delta}} G_2(j) = \bigcup_{j=b'_i}^{t'_i} G_2(j + (b'_{i+\Delta} - b'_i))$ and $(b'_{i+\Delta} - b'_i)$ is a multiple of N_{G_2}, we have $G_2 \trianglelefteq G'$, $P_{G'}^{G_2} = \frac{\Delta \cdot P_{G_1} \cdot N_{G_2}}{N_{G_1} \cdot P_{G_2}}$ and $\perp \trianglelefteq G_2$. Hence, all the hypothesis of Theorem 1 hold, and its application leads the thesis of this proposition.

Part 2 Since G_2 partitions G' (see table 2.2 of [BJW00]), then (1) $\lceil l_{G_2} \rceil_{G_2}^{G'}$ is always defined and (2) $min(\{n \in \mathbb{N}^+ | \exists i \in \mathcal{L}_{G_2} s.t. \perp(n) \subseteq G_2(i)\}) = min(\{m \in \mathbb{N}^+ | \exists j \in \mathcal{L}_{G'} s.t. \perp(m) \subseteq G'(j)\})$. Therefore $l_{G'}$ is the label of the granule of G' that covers the granule of G_2 labeled with l_{G_2}; by definition of $\lceil \cdot \rceil$ operation, $l_{G'} = \lceil l_{G_2} \rceil_{G_2}^{G'}$.

A.4 Proof of Proposition 3

Part 2 By definition of *Shift* operation, $G'(i) = G(i - m)$. Hence $G'(l_G + m) = G(l_G + m - m) = G(l_G)$.

A.5 Proof of Proposition 4

Part 1 The thesis will follow from the application of Theorem 1. Indeed, we know that $\perp \trianglelefteq G_2$ and we show that $G_2 \trianglelefteq G'$ with $P_{G'}^{G_2}$ multiple of N_{G_2}. For this we need to identify Δ and α s.t., for each i, if there exists $s(i)$ s.t. $G'(i) = \bigcup_{j \in s(i)} G_2(j)$, then $G'(i + \Delta) = \bigcup_{j \in s(i)} G_2(j + \alpha N_{G_2})$.

Consider an arbitrary $i \in \mathbb{N}$ and $\Delta = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}}$. By definition of the combining operation, we have $G'(i) = \bigcup_{j \in s(i)} G_2(j)$ and

$$G'(i + \Delta) = \bigcup_{j \in s(i+\Delta)} G_2(j)$$

with $s(i) = \{j \in \mathcal{L}_{G_2} | \emptyset \neq G_2(j) \subseteq G_1(i)\}$ and

$$s(i + \Delta) = \{j \in \mathcal{L}_{G_2} | \emptyset \neq G_2(j) \subseteq G_1(i + \Delta)\}$$

We now show that $s(i + \Delta)$ is composed by all and only the elements of $s(i)$ when the quantity $\Delta' = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_2}}{P_{G_2}}$ is added. For this purpose we need:

$$\forall j \in s(i)\ \exists (j + \Delta') \in s(i + \Delta) \tag{13}$$

and

$$\forall\, (j + \Delta') \in s(i + \Delta)\ \exists j \in s(i) \tag{14}$$

About 13, note that if $j \in s(i)$, then $G_2(j) \subseteq G_1(i)$. Since $\perp \trianglelefteq G_2$, if

$$G_2(j) = \bigcup_{r=0}^{k} \perp(j_r)$$

then

$$G_2\,(j + \Delta') = \bigcup_{r=0}^{k} \perp(j_r + lcm(P_{G_1}, P_{G_2})) \tag{15}$$

Since $G_1(i) \supseteq G_2(j) = \bigcup_{r=0}^{k} \bot(j_r)$, and since $\bot \trianglelefteq G_1$, then

$$G_1(j + \Delta) \supseteq \bigcup_{r=0}^{k} \bot(j_r + lcm(P_{G_1}, P_{G_2})) \tag{16}$$

From 15 and 16 we derive $G_1(i + \Delta) \supseteq G_2(j + \Delta')$, and hence $(j + \Delta') \in s(i+\Delta)$. Analogously can be proved the validity of 14; Hence, for each i, if there exists $s(i)$ s.t. $G'(i) = \bigcup_{j \in s(i)} G_2(j)$, then $G'(i+\Delta) = \bigcup_{j \in s(i)} G_2(j+\Delta')$. Hence, considering the fact that $G_2 \trianglelefteq G'$, we can conclude $G_2 \trianglelefteq G'$. Finally, since $P_{G'}^{G_2}$ is a multiple of N_{G_2}, by Theorem 1 we obtain the thesis.

Part 2 Let

$$\widetilde{\mathcal{L}}_{G'} = \{i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \widetilde{s}(i) \neq \emptyset\}$$

where $\forall i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ $\widetilde{s}(i) = \{j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} | \emptyset \neq G_2(j) \subseteq G_1(i)\}$;

We show that $\widetilde{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G'}$ by proving that: (1) $\widetilde{\mathcal{L}}_{G'} \supseteq \hat{\mathcal{L}}_{G'}$ and (2) $\widetilde{\mathcal{L}}_{G'} \subseteq \hat{\mathcal{L}}_{G'}$.

(1) Suppose by contradiction that exists $k \in \hat{\mathcal{L}}_{G'} \setminus \widetilde{\mathcal{L}}_{G'}$. Since $k \in \hat{\mathcal{L}}_{G'}$ and since G' is derived by the *Combine* operation, then $\exists q \in \mathcal{L}_{G_2} | G_2(q) \subseteq G_1(k)$. By definition of *Combine* operation $G'(k) = \bigcup_{j \in s(k)} G_2(j)$; since $q \in s(k)$, then $G_2(q) \subseteq G'(k)$. Hence (a) $\exists q \in \mathcal{L}_{G_2} | G_2(q) \subseteq G'(k)$.

Moreover, since $k \notin \widetilde{\mathcal{L}}_{G'}$, then $\widetilde{s}(k) = \emptyset$; therefore $\nexists \mathbf{j} \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} | G_2(j) \subseteq G_1(k)$. By definition of *Combine* operation it is easily seen that $G' \preceq G_1$. Using this and the previous formula, we derive that (b) $\nexists j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} | G_2(j) \subseteq G'(k)$.

From (a) and (b) follows that $\exists q \in \mathcal{L}_{G_2} \setminus \hat{\mathcal{L}}_{G_2}^{P_{G'}} | G_2(q) \subseteq G'(k)$. We show that this leads to a contradiction.

Since $q \notin \hat{\mathcal{L}}_{G_2}^{P_{G'}}$ and labels of $\hat{\mathcal{L}}_{G_2}^{P_{G'}}$ are contiguous (i.e., $\nexists i \in \mathcal{L}_{G_2} \setminus \hat{\mathcal{L}}_{G_2}^{P_{G'}} | min(\hat{\mathcal{L}}_{G_2}^{P_{G'}}) < i < max(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$), then $q < min(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$ or $q > max(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$. We consider the first case, the proof for the second is analogous.

If $q < min(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$ then $max(\lfloor q \rfloor^{G_2}) < 1$ (otherwise $q \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$).

Let be $\alpha = min(\lfloor min(\hat{\mathcal{L}}_{G'}) \rfloor^{G'})$. Since $k \in \hat{\mathcal{L}}_{G'}$, then $\alpha \leq \lfloor k \rfloor^{G'}$.

If $\alpha \geq 1$, then $G'(k) \cap G_2(q) = \emptyset$ contradicting $G'(k) \supseteq G_2(q)$.

If $\alpha < 1$, then $G'(l_{G'}) \supseteq \bot(0)$ and we show that $l_{G'} \in \widetilde{\mathcal{L}}_{G'}$. Indeed, by definition of *Combine*, $\exists j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} | G_2(j) \subseteq G'(L_{G'})$. Since $G' \preceq G_1$ we also have $\exists j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} | G_2(j) \subseteq G_1(L_{G'})$; hence $j \in \widetilde{s}(l_{G'})$ and then $l_{G'} \in \widetilde{\mathcal{L}}_{G'}$.

Since $0 \in G'(l_{G'})$ and $max(\lfloor q \rfloor^{G_2}) \leq 0$, then $max(\lfloor q \rfloor^{G_2}) < \alpha$ (otherwise $G_2(q) \subseteq G'(l_{G'})$). Therefore, since $min(\lfloor k \rfloor^{G'}) \geq \alpha$, then $\lfloor q \rfloor^{G_2} \cap \lfloor l_{G'} \rfloor^{G'} = \emptyset$, in contradiction with $G_2(q) \subseteq G'(k)$.

(2) Suppose by contradiction that $\exists k \in \widetilde{\mathcal{L}}_{G'} \setminus \hat{\mathcal{L}}_{G'}$. Since $k \in \widetilde{\mathcal{L}}_{G'}$, by definition of $\widetilde{\mathcal{L}}$, $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ and $\widetilde{s}(k) \neq \emptyset$; Therefore, by definition of $\widetilde{s}$, $\exists j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}} | G_2(j) \subseteq G_1(k)$.

Since $j \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$, by definition of $\hat{\mathcal{L}}$, $\exists h$ with $0 < h \leq P_{G'}$ s.t. $\lceil h \rceil^{G_2} = j$. Since $G_2(j) \subseteq G_1(k)$, then $\lceil h \rceil^{G_1} = k$. By definition of *combine* operation, $\lceil h \rceil^{G'} = k$.

Moreover, since $0 < h \leq P_{G'}$, by definition of $\hat{\mathcal{L}}$, $\lceil h \rceil^{G'} = k \in \hat{\mathcal{L}}_{G'}$, contradicting the hypothesis.

A.6 Proof of Proposition 5

Part 1 The thesis will follow from the application of Theorem 1. Indeed, we show that $G_1 \trianglelefteq G'$ with $P_{G'}^{G_1}$ multiple of N_{G_1}. For this we need to identify Δ and α s.t., for each i, if there exists $s(i)$ s.t. $G'(i) = \bigcup_{j \in s(i)} G_1(j)$, then $G'(i+\Delta) = \bigcup_{j \in s(i)} G_1(j + \alpha N_{G_1})$. Let $\Delta = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_2}}{P_{G_2}}$. By definition of anchored grouping, $G'(i) = \bigcup_{j=i}^{i'-1} G_1(j)$ and $G'(i+\Delta) = \bigcup_{j=i+\Delta}^{(i+\Delta)'-1}$ where i' is the first label of G_2 after i and $(i+\Delta)'$ is the first label of G_2 after $i+\Delta$. By periodicity of G_2, (and since Δ is a multiple of N_{G_2}) the difference between the label of the granule following $G_2(i+\Delta)$ and the label of the granule following $G_2(i)$ is Δ. More formally, $(i+\Delta)' - i' = \Delta$, hence $(i+\Delta)' = i' + \Delta$. Then, for each i, if $G'(i) = \bigcup_{j=i}^{k} G_1(j)$, then $G'(i+\Delta) = \bigcup_{j=i+\Delta}^{i'+\Delta-1} G_1(j) = \bigcup_{j=i}^{i'-1} G_1(j+\Delta)$. By this result and considering $G_1 \trianglelefteq G'$, we conclude $G_1 \trianglelefteq G'$ with $P_{G'}^{G_1} = \Delta$. Note that by Proposition 9, $N_{G_1} = \frac{P_{G_1} \cdot N_{G_2}}{P_{G_2}}$, hence $P_{G'}^{G_1}$ is a multiple of Δ. Then, by Theorem 1, we have the thesis.

Part 2 Let

$$\widetilde{\mathcal{L}}_{G'} = \begin{cases} \hat{\mathcal{L}}_{G_2}^{P_{G'}}, & \text{if } l_{G_2} = l_{G_1}, \\ \{l'_{G_2}\} \cup \hat{\mathcal{L}}_{G_2}^{P_{G'}}, & \text{otherwise,} \end{cases}$$

where l'_{G_2} is the greatest among the labels of $\mathcal{L}_{G_2}$ that are smaller than l_{G_2}. We show that $\widetilde{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G'}$ by proving that (1) $\widetilde{\mathcal{L}}_{G'} \subseteq \hat{\mathcal{L}}_{G'}$ and (2) $\hat{\mathcal{L}}_{G'} \subseteq \widetilde{\mathcal{L}}_{G'}$.

(1) Suppose by contradiction that $\exists k \in \widetilde{\mathcal{L}}_{G'} \setminus \hat{\mathcal{L}}_{G'}$. Then, since $k \in \widetilde{\mathcal{L}}_{G'}$, then $k \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$ or $k = l'_{G_2}$.

If $k \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$, then, by definition of $\hat{\mathcal{L}}_{G_2}^{P_{G'}}$, $\exists h$ with $0 < h \leq P_{G'}$ s.t. $\lceil h \rceil^{G_2} = k$. By definition of *Anchored-group*, $G'(k) = \bigcup_{j=k}^{k'-1} G_1(j)$ where k' is the first label of G_2 after k. Therefore $G'(k) \supseteq G_1(k)$. Since G_2 is a labeled aligned subgranularity of G_1 and since $k \in \mathcal{L}_{G_2}$, then $k \in \mathcal{L}_{G_1}$ and $G_1(k) = G_2(k)$. Hence $G'(k) \supseteq G_2(k)$. Follows that $\lceil h \rceil^{G'} = k$ and therefore, by definition of $\hat{\mathcal{L}}$, $k \in \hat{\mathcal{L}}_{G'}$ in contrast with the hypothesis.

If $k = l'_{G_2}$, then, by definition of $\widetilde{\mathcal{L}}_{G'}$, $l_{G_2} \neq l_{G_1}$. Therefore, since G_2 is a labeled aligned subgranularity of G_1 $l'_{G_2} < l_{G_1} < l_{G_2}$; then $\exists h$ with $0 < h < min(\lfloor l_{G_2} \rfloor^{G_2})$ s.t. $\lceil h \rceil^{G_1} = l_{G_1}$. Since, by definition of *Anchored-group*, $G'(l'_{G_2}) = \bigcup_{j=l'_{G_2}}^{l_{G_2}-1} G_1(j)$ and since $l'_{G_2} < l_{G_1} < l_{G_2}$, then $G'(l'_{G_2}) \supseteq G_1(l_{G_1})$. Hence $\lceil h \rceil^{G'} = l'_{G_2}$ and therefore, by definition of $\hat{\mathcal{L}}$, $l'_{G_2} = k \in \hat{\mathcal{L}}_{G'}$ in contrast with the hypothesis.

(2) Suppose by contradiction that $\exists k \in \hat{\mathcal{L}}_{G'} \setminus \widetilde{\mathcal{L}}_{G'}$. If $k \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$ then, by definition of $\widetilde{\mathcal{L}}_{G'}$, $k \in \widetilde{\mathcal{L}}_{G'}$, in contrast with the hypothesis.

If $k \notin \hat{\mathcal{L}}_{G_2}^{P_{G'}}$, since $\nexists q \in \mathcal{L}_{G_2} \setminus \hat{\mathcal{L}}_{G_2}^{P_{G'}}$ s.t. $min(\hat{\mathcal{L}}_{G_2}^{P_{G'}}) \leq q \leq max(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$, then $k > max(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$ or $k < min(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$.

If $k > max(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$ then, by definition of $\hat{\mathcal{L}}$, $min(\lfloor k \rfloor^{G_2}) > P_{G'}$. Since G_2 is a labeled aligned subgranularity of G_1 then $G_2(k) = G_1(k)$ and hence $min(\lfloor k \rfloor^{G_1}) > P_{G'}$. Since $G'(k) = \bigcup_{j=k}^{k'-1} G_1(j)$ then $min(\lfloor k \rfloor^{G'}) > P_{G'}$ in contrast with the hypothesis $k \in \hat{\mathcal{L}}_{G'}$.

If $k < min(\hat{\mathcal{L}}_{G_2}^{P_{G'}})$ then, by definition of l'_{G_2}, $k < l'_{G_2}$ or $k = l'_{G_2}$.

If $k < l'_{G_2}$ then, let k' be the next label of G_2 after k. Since $k < l'_{G_2}$ then, by definition l'_{G_2}, $k' \leq l'_{G_2}$. By definition of l'_{G_2} then $max(\lfloor l'_{G_2} \rfloor^{G_2}) \leq 0$. Since G_2 is a labeled aligned subgranularity of G_1 then $G_1(l'_{G_2}) = G_2(l'_{G_2})$; therefore $max(\lfloor l'_{G_2} \rfloor^{G_1}) \leq 0$. Since $G'(k) = \bigcup_{j=k}^{k'-1} G_1(j)$ and $k' \leq l'_{G_2}$, follows that $max(\lfloor k \rfloor^{G'}) \leq 0$ in contrast with the hypothesis $k \in \hat{\mathcal{L}}_{G'}$.

Finally if $k = l'_{G_2}$ then $G'(l'_{G_2}) = \bigcup_{j=l'_{G_2}}^{l_{G_2}-1} G_1(j)$. Since $k = l'_{G_2} \in \hat{\mathcal{L}}_{G'}$ then $\exists h$ with $0 < h \leq P_{G'}$ s.t. $\lceil h \rceil^{G'} = l'_{G_2}$. Since G' is the composition of granules of G_1, $\lceil h \rceil^{G_1}$ is defined. Let $q = \lceil h \rceil^{G_1}$. By definition of $\hat{\mathcal{L}}$, $q \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ and therefore $q \geq l_{G_1}$. Since, by definition of *Anchored-group*, G' is the composition of granules of G_1 and since $\lceil h \rceil^{G'} = l'_{G_2}$ and $\lceil h \rceil^{G_1} = q$, then $G_1(q) \subseteq G'(l'_{G_2})$. Therefore since $G'(l'_{G_2}) = \bigcup_{j=l'_{G_2}}^{l_{G_2}-1} G_1(j)$ then $q < l_{G_2}$. Follows that $l_{G_1} \leq q < l_{G_2}$ and hence $l_{G_1} \neq l_{G_2}$. By definition of $\widetilde{\mathcal{L}}_{G'}$, $l'_{G_2} = k \in \widetilde{\mathcal{L}}_{G'}$ in contrast with the hypothesis.

A.7 Selecting operations

The selecting operations has a common part in the proof for the computation of the period and the period label distance.

Let be $\Gamma = \frac{lcm(P_{G_1}, P_{G_2}) N_{G_1}}{P_{G_1}}$. The proof is divided into two steps: first we show that for each select operation if $i \in \mathcal{L}_{G'}$ then $i + \Gamma \in \mathcal{L}_{G'}$ (details for *Select-down*, *Select-up* and *Select-by-intersect* operations can be found below). The second step is the application of Theorem 1. Indeed, for each *Select* operation, the following holds: $\forall i \in \mathcal{L}_{G'}\ G'(i) = G_1(i)$; this implies $G_1 \trianglelefteq G'$. From step 1 follows that $i + \Gamma \in \mathcal{L}_{G'}$, hence $G'(i + \Gamma) = G_1(i + \Gamma)$. By this result and considering $G_1 \trianglelefteq G'$, we conclude that $G_1 \bar{\trianglelefteq} G'$ with $P_{G'}^{G_1} = \Gamma$ which is a multiple of N_{G_1} by definition. Then, by Theorem 1 we have the thesis.

A.8 Proof of Proposition 6

Part 1 See Section A.7.

We prove that if $\lambda \in \mathcal{L}_{G'}$ then $\lambda' = \lambda + \Gamma \in \mathcal{L}_{G'}$.

By definition of the `select-down` operation, if $\lambda \in \mathcal{L}_{G'}$ then $\exists i \in \mathcal{L}_{G_2}$ s.t. $\lambda \in \Delta_k^l(S(i))$ where $S(i)$ is an ordered set defined as follows: $S(i) = \{j \in \mathcal{L}_{G_1} | \emptyset \neq G_1(j) \subseteq G_2(i)\}$. In order to prove the thesis we need to show that

$\exists i' \in \mathcal{L}_{G_2} | \lambda' \in \Delta_k^l(S(i'))$. Consider $i' = i + \frac{lcm(P_{G_1} P_{G_2}) N_{G_2}}{P_{G_2}}$ we will note that $i' \in \mathcal{L}_{G_2}$ (this is trivially derived from the periodicity of G_2). To prove that $\lambda' \in \Delta_k^l(S(i'))$ we show that $S(i')$ is obtained from $S(i)$ by adding Γ to each of its elements.

Indeed note that from periodicity of G_1, $\forall j \in S(i)$ if:

$$G_1(j) = \bigcup_{r=0}^{\tau_j} \bot(j_r) \tag{17}$$

then:

$$G_1(j') = \bigcup_{r=0}^{\tau_j} \bot(j_r + lcm(P_{G_1} P_{G_2})) \tag{18}$$

Since $j \in S(i)$, $G_1(j) \subseteq G_2(i)$ then, from (17), $G_2(i) \supseteq \bigcup_{r=0}^{\tau_j} \bot(j_r)$. Moreover, from periodicity of G_2:

$$G_2(i') \supseteq \bigcup_{r=0}^{\tau_j} \bot(j_r + lcm(P_{G_1} P_{G_2})) \tag{19}$$

Since (18) and (19), $G_2(i') \supseteq G_1(j')$; hence $\forall j \in S(i), j' = (j + \Gamma) \in S(i')$. Analogously we can prove that $\forall j' \in S(i'), j = (j' - \Gamma) \in S(i)$.

Thus $S(i')$ is obtained from $S(i)$ by adding Γ to each of its elements; therefore if $j \in S(i)$ has position n in $S(i)$, so $j' \in S(i')$ has position n in $S(i')$. Hence it is trivial that if λ has position between k and $k+l-1$ in $S(i)$, then λ' has position between k and $k+l-1$ in $S(i')$. Hence if $\lambda \in \mathcal{L}_{G'}$, then $\lambda' \in \mathcal{L}_{G'}$.

Part 2 Let

$$\widetilde{\mathcal{L}}_{G'} = \bigcup_{i \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}} \left\{ a \in A(i) | a \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} \right\};$$

where $\forall i \in \mathcal{L}_{G_2}$:

$$A(i) = \Delta_k^l \left(\{ j \in \mathcal{L}_{G_1} | \emptyset \neq G_1(j) \subseteq G_2(i) \} \right).$$

We show that $\widetilde{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G'}$ by proving that (1) $\widetilde{\mathcal{L}}_{G'} \subseteq \hat{\mathcal{L}}_{G'}$ and (2) $\widetilde{\mathcal{L}}_{G'} \supseteq \hat{\mathcal{L}}_{G'}$.

(1)Suppose by contradiction that $\exists q \in \widetilde{\mathcal{L}}_{G'} \setminus \hat{\mathcal{L}}_{G'}$. By definition of $\widetilde{\mathcal{L}}_{G'}$, $q \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$; therefore $\exists h$ with $0 < h \leq P_{G'}$ s.t. $\lceil h \rceil^{G_1} = q$. Moreover, by definition of $\widetilde{\mathcal{L}}_{G'}$ and by definition of *Select-down*, $\widetilde{\mathcal{L}}_{G'} \subseteq \mathcal{L}_{G'}$ hence $q \in \mathcal{L}_{G'}$. Since, by definition of *Select-down* $G'(q) = G_1(q)$, then $\lceil h \rceil^{G'} = q$; hence, by definition of $\hat{\mathcal{L}}$, $q \in \hat{\mathcal{L}}_{G'}$ in contradiction with hypothesis.

(2)Suppose by contradiction that $\exists q \in \hat{\mathcal{L}}_{G'} \setminus \widetilde{\mathcal{L}}_{G'}$. Since $q \in \hat{\mathcal{L}}_{G'}$ then, by definition of *Select-down*

$$\exists i \in \mathcal{L}_{G_2} \; s.t. \; q \in \Delta_k^l \left(\{ j \in \mathcal{L}_{G_1} | \emptyset \neq G_1(j) \subseteq G_2(i) \} \right)$$

therefore, by definition of $A(i)$, $q \in A(i)$.

Since $q \in \hat{\mathcal{L}}_{G'}$ then $\exists h$ with $0 < h \leq P_{G'}$ s.t. $\lceil h \rceil^{G'} = q$. By definition of *Select-down*, $G'(q) = G_1(q)$, then $\lceil h \rceil^{G_1} = q$ and therefore $q \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$. Moreover, since $G_1(q) \subseteq G_2(i)$, then $\lceil h \rceil^{G_2} = i$ and therefore $i \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$. Since $q \in A(i)$, $q \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ and $i \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$ then, by definition of $\widetilde{\mathcal{L}}_{G'}$, $q \in \widetilde{\mathcal{L}}_{G'}$, in contrast with the hypothesis.

A.9 Proof of Proposition 7

Part 1 See Section A.7. We prove that if $i \in \mathcal{L}_{G'}$ then $i + \Gamma \in \mathcal{L}_{G'}$. From the periodicity of G_1, $i + \Gamma \in \mathcal{L}_{G_1}$ (this is trivially derived from the periodicity of G_1). Hence we only need to show that $\exists j' \in \mathcal{L}_{G_2} | \emptyset \neq G_2(j) \subseteq G_1(i + \Gamma)$. Since $i \in \mathcal{L}_{G'}$ then $\exists j \in \mathcal{L}_{G_2} | \emptyset \neq G_2(j) \subseteq G_1(i)$.

From the periodicity of G_2, if:

$$G_2(j) = \bigcup_{r=0}^{\tau_j} \perp(j_r) \tag{20}$$

then:

$$G_2\left(j + \frac{lcm(P_{G_1}P_{G_2})N_{G_2}}{P_{G_2}}\right) = \bigcup_{r=0}^{\tau_j} \perp(j_r + lcm(P_{G_1}P_{G_2})) \tag{21}$$

Moreover, from the (20) and since $G_1(i) \supseteq G_2(j)$:

$$G_1(i) \supseteq \bigcup_{r=0}^{\tau_j} \perp(j_r)$$

From the periodicity of G_1:

$$G_1(i + \Gamma) \supseteq \bigcup_{r=0}^{\tau_j} \perp(j_r + lcm(P_{G_1}P_{G_2})) \tag{22}$$

From (21) and (22) follows that $G_1(i + \Gamma) \supseteq G_2\left(j + \frac{lcm(P_{G_1}P_{G_2})N_{G_2}}{P_{G_2}}\right)$, that is the thesis.

Part 2 let

$$\widetilde{\mathcal{L}}_{G'} = \{i \in \hat{\mathcal{L}}_{G_1}^{P_{G'}} | \exists j \in \mathcal{L}_{G_2} \; s.t. \, \emptyset \neq G_2(j) \subseteq G_1(i)\};$$

We show that $\widetilde{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G'}$ by proving that (1) $\widetilde{\mathcal{L}}_{G'} \subseteq \hat{\mathcal{L}}_{G'}$ and (2) $\widetilde{\mathcal{L}}_{G'} \supseteq \hat{\mathcal{L}}_{G'}$.

(1) Suppose by contradiction that $\exists k \in \widetilde{\mathcal{L}}_{G'} \setminus \hat{\mathcal{L}}_{G_2}$. Since $k \in \widetilde{\mathcal{L}}_{G'}$, then $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$; therefore $\exists h$ with $0 < h \leq P_{G'}$ s. t. $\lceil h \rceil^{G_1} = k$. Moreover, by definition

of $\widetilde{\mathcal{L}}_{G'}$ and by definition of *Select-down*, $\widetilde{\mathcal{L}}_{G'} \subseteq \mathcal{L}_{G'}$ hence $q \in \mathcal{L}_{G'}$. Since, by definition of *Select-up*, $G'(k) = G_1(k)$, then $\lceil h \rceil^{G'} = k$. Hence, by definition of $\hat{\mathcal{L}}$, $k \in \hat{\mathcal{L}}_{G'}$, in contrast with the hypothesis.

(2) Suppose by contradiction that $\exists k \in \hat{\mathcal{L}}_{G'} \setminus \widetilde{\mathcal{L}}_{G'}$. Since $k \in \hat{\mathcal{L}}_{G'}$, then $\exists h$ with $0 < h \leq P_{G'}$ s.t. $\lceil h \rceil^{G'} = k$. Since, by definition of *Select-up*, $G'(k) = G_1(k)$, then $\lceil h \rceil^{G_1} = k$; Therefore, by definition of $\hat{\mathcal{L}}$, $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$. Moreover, since $k \in \hat{\mathcal{L}}_{G'}$ and $\hat{\mathcal{L}}_{G'} \subseteq \mathcal{L}_{G'}$, by definition of *Select-up* operation, then $\exists j \in \mathcal{L}_{G_2}$ s.t. $\emptyset \neq G_2(j) \subseteq G_1(k)$. Hence by definition of $\widetilde{\mathcal{L}}_{G'}$, $k \in \widetilde{\mathcal{L}}_{G'}$, in contradiction with hypothesis.

A.10 Proof of Proposition 8

Part 1 See Section A.7. We prove that if $\lambda \in \mathcal{L}_{G'}$, then $\lambda' = \lambda + \Gamma \in \mathcal{L}_{G'}$.

By definition of the `select-by-intersect` operation, if $\lambda \in \mathcal{L}_{G'}$, then $\exists i \in \mathcal{L}_{G_2} : \lambda \in \Delta_k^l(S(i))$ where $S(i)$ is an ordered set defined as follows: $S(i) = \{j \in \mathcal{L}_{G_1} | G_1(j) \cap G_2(i) \neq \emptyset\}$. In order to prove the thesis we need to show that $\exists i' \in \mathcal{L}_{G_2} : \lambda' \in \Delta_k^l(S(i'))$. Consider $i' = i + \frac{lcm(P_{G_1} P_{G_2}) N_{G_2}}{P_{G_2}}$ note that $i' \in \mathcal{L}_{G_2}$ (this is trivially derived from the periodicity of G_2). To prove that $\lambda' \in \Delta_k^l(S(i'))$ we show that $S(i')$ is obtained from $S(i)$ by adding Γ to each of its elements.

Indeed note that $\forall j$ if $j \in S(i)$, then $G_1(j) \cap G_2(i) \neq \emptyset$. Hence $\exists l \in \mathbb{Z} : \perp(l) \subseteq G_1(j)$ and $\perp(l) \subseteq G_2(i)$. From the periodicity of G_1, $G_1(j + \Gamma) \supseteq \perp(l + lcm(P_{G_1} P_{G_2}))$. From the periodicity of G_2, $G_2(i') \supseteq \perp(l + lcm(P_{G_1} P_{G_2}))$. So $G_1(j + \Gamma) \cap G_2(i') \neq \emptyset$, therefore $\forall j \in S(i), (j + \Gamma) \in S(i')$.

Analogously we can prove that $\forall j' \in S(i'), (j' - \Gamma) \in S(i)$. Hence $S(i')$ is obtained from $S(i)$ by adding Γ to each of its elements. Therefore, if $j \in S(i)$ has position n in $S(i)$, then $j + \Gamma \in S(i')$ has position n in $S(i')$; hence if j has position between k and $k + l - 1$ in $S(i)$, then also $j + \Gamma$ has position between k and $k + l - 1$ in $S(i')$ and so $j + \Gamma \in \mathcal{L}_{G'}$.

Part 2 The proof is analogous to the ones of Proposition 6.

A.11 Set Operations

Proof of proposition 9 Given the periodical granularities H and G with G labeled aligned subgranularity of H, we prove that $\frac{N_G}{P_G} = \frac{N_H}{P_H}$. The thesis is proved by considering the common period of H and G i.e. $P_c = lcm(P_G, P_H)$.

Let N'_G be the difference between the label of the i^{th} granule of one period of G and the label of the i^{th} granule of the next period, considering P_c as the period of G. Analogously N'_H is defined.

By periodicity of G, if $G(i) = \bigcup_{r=0}^{k} \perp(i_r)$ then $G(i + N'_G) = \bigcup_{r=0}^{k} \perp(i_r + P_c)$; since G is an aligned subranularity of H, $\forall i \in \mathcal{L}_H \ H(i) = G(i) = \bigcup_{r=0}^{k} \perp(i_j)$ and, since H is periodic, $H(i + N'_H) = \bigcup_{r=0}^{k} \perp(i_j + P_c)$; from which we can easily derive that $i + N'_G = i + N'_H$, hence $N'_G = N'_H$.

From the definition of P_c, $\exists \alpha, \beta \in \mathbb{N}$ s. t. $\alpha P_H = \beta P_G$. Moreover, since $N'_H = N'_G$, then $\alpha N_H = \beta N_G$. Therefore $\frac{P_H}{N_H} = \frac{P_G}{N_G}$.

Property used for the proofs of set operations Let $\mathit{\Gamma}_1$ be $\frac{lcm(P_{G_1},P_{G_2})N_{G_1}}{P_{G_1}}$ and $\mathit{\Gamma}_2$ be $\frac{lcm(P_{G_1},P_{G_2})N_{G_2}}{P_{G_2}}$. Since G_1 and G_2 are aligned subgranularity of a certain granularity H, from proposition 9 we can easily derive that $\mathit{\Gamma}_1 = \mathit{\Gamma}_2$.

A.12 Proof of Proposition 10

Part 1 The thesis will be proved by showing that $\forall i$ if, $G'(i) = \bigcup_{r=0}^{k} \perp(i_r)$, then $G'(i + \mathit{\Delta}) = \bigcup_{r=0}^{k} \perp(i_r + lcm(P_{G_1}, P_{G_2}))$ with $\mathit{\Delta} = \mathit{\Gamma}_1 = \mathit{\Gamma}_2$. Since $\mathcal{L}_{G'} = \mathcal{L}_{G_1} \cup \mathcal{L}_{G_2}$, two cases will be considered:

- $\forall i \in \mathcal{L}_{G_1}$ $G'(i) = G_1(i) = \bigcup_{r=0}^{k} \perp(i_r)$. From the periodicity of G_1, $G_1(i + \mathit{\Gamma}_1) = \bigcup_{r=0}^{k} \perp(i_r + lcm(P_{G_1}, P_{G_2}))$; hence $G'(i+\mathit{\Gamma}_1) = \bigcup_{r=0}^{k} \perp(i_r + lcm(P_{G_1}, P_{G_2}))$.
- $\forall i \in \mathcal{L}_{G_2} - \mathcal{L}_{G_1}$ $G'(i) = G_2(i) = \bigcup_{r=0}^{k} \perp(i_r)$. From the periodicity of G_2, $G_2(i + \mathit{\Gamma}_2) = \bigcup_{r=0}^{k} \perp(i_r + lcm(P_{G_1}, P_{G_2}))$; hence $G'(i + \mathit{\Gamma}_2) = \bigcup_{r=0}^{k} \perp(i_r + lcm(P_{G_1}, P_{G_2}))$.

Since $\mathit{\Gamma}_1 = \mathit{\Gamma}_2$, then $\forall i \in \mathcal{L}_{G'}$ if $G'(i) = \bigcup_{r=0}^{k} \perp(i_r)$, then $G'(i + \mathit{\Gamma}_1) = G'(i + \mathit{\Gamma}_2) = \bigcup_{r=0}^{k} \perp(i_r + lcm(P_{G_1}, P_{G_2}))$. Hence, by definition of $\trianglelefteq$, we have the thesis.

Part 2 Let $\widetilde{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G_1}^{P_{G'}} \cup \hat{\mathcal{L}}_{G_2}^{P_{G'}}$.

We show that $\widetilde{\mathcal{L}}_{G'} = \hat{\mathcal{L}}_{G'}$ by proving that (1) $\widetilde{\mathcal{L}}_{G'} \subseteq \hat{\mathcal{L}}_{G'}$ and (2) $\widetilde{\mathcal{L}}_{G'} \supseteq \hat{\mathcal{L}}_{G'}$.

(1) Suppose by contradiction that $\exists k \in \widetilde{\mathcal{L}}_{G'} \setminus \hat{\mathcal{L}}_{G'}$. Since $k \in \widetilde{\mathcal{L}}_{G'}$ then $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ or $k \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$. Suppose that $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ (the proof is analogous if $k \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$). Since $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$, then $\exists\, 0 < h < P_{G'}$ s.t. $\lceil h \rceil^{G'} = k$. Since, by definition of *Union* operation $G'(k) = G_1(k)$, then $\lceil h \rceil^{G'} = k$. Hence, by definition of $\hat{\mathcal{L}}$, $k \in \hat{\mathcal{L}}_{G'}$ in contrast with the hypothesis.

(2) Suppose by contradiction that $\exists k \in \hat{\mathcal{L}}_{G'} \setminus \widetilde{\mathcal{L}}_{G'}$. Since $k \in \hat{\mathcal{L}}_{G'}$, then, by definition of $\hat{\mathcal{L}}$, $\exists\, 0 < h < P_{G'}$ s.t. $\lceil h \rceil^{G'} = k$. Moreover, by definition of *Union* operation, $k \in \mathcal{L}_{G_1}$ or $k \in \mathcal{L}_{G_2}$. Suppose that $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$ (the proof is analogous if $k \in \hat{\mathcal{L}}_{G_2}^{P_{G'}}$). By definition of *Union* operation, $G'(k) = G_1(k)$ therefore $\lceil h \rceil^{G_1} = k$ and so, by definition of $\hat{\mathcal{L}}$, $k \in \hat{\mathcal{L}}_{G_1}^{P_{G'}}$. Hence, by definition of $\widetilde{\mathcal{L}}$, $k \in \widetilde{\mathcal{L}}_{G'}$ in contradiction with the hypothesis.

A.13 Proof of Proposition 11

Part 1 $\forall i \in \mathcal{L}_{G'} = \mathcal{L}_{G_1} \cap \mathcal{L}_{G_2}$ $G'(i) = G_1(i) = \bigcup_{r=0}^{k} \perp(i_r)$. From the periodicity of G_1 and G_2, $i + \mathit{\Gamma}_1 \in \mathcal{L}_{G_1}$ e $i + \mathit{\Gamma}_2 \in \mathcal{L}_{G_2}$; since $\mathit{\Gamma}_1 = \mathit{\Gamma}_2$, then $i + \mathit{\Gamma}_1 \in \mathcal{L}_{G'}$. Moreover $G'(i + \mathit{\Gamma}_1) = G_1(i + \mathit{\Gamma}_1) = \bigcup_{r=0}^{k} \perp(i_r + lcm(P_{G_1}, P_{G_2}))$; hence, by the definition of $\trianglelefteq$, we have the thesis.

Part 2 The proof is analogous to the one of Proposition 10.

A.14 Proof of Proposition 12

Part 1 $\forall i \in \mathcal{L}_{G'} = \mathcal{L}_{G_1} - \mathcal{L}_{G_2}$ $G'(i) = G_1(i) = \bigcup_{r=0}^{k} \bot(i_r)$. Since $i \in \mathcal{L}_{G_1}$ from the periodicity of G_1 $i + \Gamma_1 \in \mathcal{L}_{G_1}$. Since $i \notin \mathcal{L}_{G_2}$, from the periodicity of G_2, $i + \Gamma_2 \notin \mathcal{L}_{G_2}$ (if it would exists $i + \Gamma_2 \in \mathcal{L}_{G_2}$, from periodicity of G_2 would exists $i \in \mathcal{L}_{G_2}$ that is not possible for hypothesis). Hence $i + \Gamma_1 \in \mathcal{L}_{G'}$. Moreover $G'(i + \Gamma_1) = G_1(i + \Gamma_1) = \bigcup_{r=0}^{k} \bot(i_r + lcm(P_{G_1}, P_{G_2}))$; hence, by the definition of $\trianglelefteq$, we have the thesis.

Part 2 The proof is analogous to the one of Proposition 10.

A.15 Proof of Lemma 1

Proof. Let $\bar{G}^1$ and $\bar{G}^2$ be two arbitrary representation of G. Moreover let $\bar{G}^3$ be the periodical representation of G s.t. $P_{\bar{G}^3} = lcm(P_{\bar{G}^1}, P_{\bar{G}^2})$; hence $\exists \alpha, \beta \in \mathbb{N}^+$ s.t. $P_{\bar{G}^3} = \alpha P_{\bar{G}^1}$ and $P_{\bar{G}^3} = \beta P_{\bar{G}^2}$. From the periodicity of $\bar{G}^3$ follows that if $G(i) = \bigcup_{k=1}^{t} \bot(j_k)$ then

$$G(i + N_{\bar{G}^3}) = \bigcup_{k=1}^{t} \bot(j_k + P_{\bar{G}^3}) = \bigcup_{k=1}^{t} \bot(j_k + \alpha P_{\bar{G}^1}) \tag{23}$$

From the periodicity of $P_{\bar{G}^1}$ and since, by assumption, $G(i) = \bigcup_{k=1}^{t} \bot(j_k)$, then $\bigcup_{k=1}^{t} \bot(j_k + \alpha P_{\bar{G}^1}) = G(i + \alpha N_{\bar{G}^1})$. Therefore from (23), $G(i + N_{\bar{G}^3}) = G(i + \alpha N_{\bar{G}^1})$ and hence $N_{\bar{G}^3} = \alpha N_{\bar{G}^1}$; Analogously $N_{\bar{G}^3} = \beta N_{\bar{G}^2}$. Hence let be $\frac{P_{\bar{G}^3}}{N_{\bar{G}^3}} = \lambda$, then $\lambda = \frac{\alpha P_{\bar{G}^1}}{\alpha N_{\bar{G}^1}} = \frac{\beta P_{\bar{G}^2}}{\beta N_{\bar{G}^2}}$; follows that $\frac{P_{\bar{G}^1}}{N_{\bar{G}^1}} = \frac{P_{\bar{G}^2}}{N_{\bar{G}^2}} = \lambda$.

Since $P_{\bar{G}^3} = \alpha P_{\bar{G}^1}$ then clearly $R_{\bar{G}^3} = \alpha R_{\bar{G}^1}$; Analogously, $R_{\bar{G}^3} = \beta R_{\bar{G}^2}$. Hence let be $\frac{P_{\bar{G}^3}}{R_{\bar{G}^3}} = \lambda'$, then $\lambda' = \frac{\alpha P_{\bar{G}^1}}{\alpha R_{\bar{G}^1}} = \frac{\beta P_{\bar{G}^2}}{\beta R_{\bar{G}^2}}$; follows that $\frac{P_{\bar{G}^1}}{R_{\bar{G}^1}} = \frac{P_{\bar{G}^2}}{R_{\bar{G}^2}} = \lambda'$.

A.16 Proof of Lemma 2

Proof. We prove (ii), then (i) and (iii) follows from lemma 1. Since $\bar{G}^1$ is minimal, then clearly $N_{\bar{G}^1} \leq N_{\bar{G}^2}$; hence $\exists \gamma, \delta \in \mathbb{N}$ s.t. $N_{\bar{G}^2} = \delta + \gamma N_{\bar{G}^1}$ with $0 \leq \delta < N_{\bar{G}^1}$ and $\gamma > 0$. Suppose $\delta > 0$; From the periodicity of $\bar{G}^1$ follows that $\forall i \in \mathcal{L}_G$ if $G(i) = \bigcup_{k=0}^{n} \bot(j_k)$ then $G(i - \gamma N_{\bar{G}^1}) = \bigcup_{k=0}^{n} \bot(j_k - \gamma P_{\bar{G}^1})$. From the periodicity of $\bar{G}^2$ follows that $\forall i \in \mathcal{L}_G$ if $G(i - \gamma N_{\bar{G}^1}) = \bigcup_{k=0}^{n} \bot(j_k - \gamma P_{\bar{G}^1})$ then $G(i - \gamma N_{\bar{G}^1} + N_{\bar{G}^2}) = \bigcup_{k=0}^{n} \bot(j_k - \gamma P_{\bar{G}^1} + P_{\bar{G}^2})$. Since $N_{\bar{G}^1} - \gamma N_{\bar{G}^1} = \delta$ then $\forall i \in \mathcal{L}_G$ if $G(i) = \bigcup_{k=0}^{n} \bot(j_k)$ then $G(i + \delta) = \bigcup_{k=0}^{n} \bot(j_k - \gamma P_{\bar{G}^1} + P_{\bar{G}^2})$. Therefore, by definition of the periodically groups into relation, there exists a periodical representation of G that has δ as the period label distance. This leads to a contradiction, in fact $\bar{G}^1$ is minimal and hence cannot exists a periodical representation of G having a smaller period label distance. It follows that $\delta = 0$; therefore $\gamma N_{\bar{G}^1} = N_{\bar{G}^2}$.

A.17 Proof of lemma 3

In this proof we indicate with (1) the condition of Section 5.1. Given $\forall i \in \mathcal{L}_G$:

$$(i+N) \in \mathcal{L}_G \text{ and if } G(i) = \bigcup_{j\in S} \bot(j), \text{then } G(i+N) = \bigcup_{j\in S} \bot(j+P) \quad (24)$$

we prove that (24)$\Leftrightarrow$(1).

$\Rightarrow$) Suppose that $\lceil k \rceil^G$ is undefined and, by contradiction, that $\lceil k+P \rceil^G$ is defined. Since $\lceil k+P \rceil^G$ is defined, then exists a set of integers S s.t. $G(\lceil k + P \rceil^G) = \bigcup_{j\in S} \bot(j+P)$; by definition of $\lceil\cdot\rceil$ operation, $k \in S$. From (24) easily follows that:

$$\forall i \in \mathcal{L}_G \ , \ (i-N) \in \mathcal{L}_G \text{ and if } G(i) = \bigcup_{j\in S} \bot(j+P), \text{then } G(i-N) = \bigcup_{j\in S} \bot(j)$$

therefore $G(\lceil k+P \rceil^G - N) = \bigcup_{j\in S} \bot(j)$; since $k \in S$, then $\lceil k \rceil^G = \lceil k+P \rceil^G - N$ and hence $\lceil k \rceil^G$ is defined, in contrast with the hypothesis.

Suppose that $\lceil k \rceil^G$ is defined, then there exists a set S of integers s.t. $G(\lceil k \rceil^G) = \bigcup_{j\in S} \bot(j)$ and, by definition of $\lceil\cdot\rceil$ operation, $k \in S$. From (24) follows that $G(\lceil k \rceil^G + N) = \bigcup_{j\in S} \bot(j+P)$; Hence, since $k \in S$, we have $\lceil k+P \rceil^G = \lceil k \rceil^G + N$ as specified in (1).

$\Leftarrow$)Given $\forall i \in \overline{\mathcal{L}}_{\bar{G}^1}$:

$$(i+N) \in \mathcal{L}_G \text{ and if } G(i) = \bigcup_{j\in S} \bot(j), \text{then } G(i+N) = \bigcup_{j\in S} \bot(j+P) \quad (25)$$

we show that (25) follows from (1) and that (24) follows from (25) [5].

(1) $\rightarrow$ (25). We prove that if (1) holds, then, for each $i \in \overline{\mathcal{L}}_{\bar{G}^1}$, $(i+N) \in \mathcal{L}_G$ and if $G(i) = \bigcup_{j\in S} \bot(j)$ then $G(i+N) = \bigcup_{j\in S} \bot(j+P)$.

Since $i \in \overline{\mathcal{L}}_{\bar{G}^1}$, by definition of K, $\exists k \in K$ s.t. $\lceil k \rceil^G = i$. By hypothesis, it follows that $\lceil k+P \rceil^G$ is defined, therefore $\lceil k+P \rceil^G \in \mathcal{L}_G$. Since, by hypothesis, $\lceil k+P \rceil^G = N + \lceil k \rceil^G$, then $(N + \lceil k \rceil^G) \in \mathcal{L}_G$, hence $(N+i) \in \mathcal{L}_G$.

If $G(i) = \bigcup_{j\in S} \bot(j)$, we prove that $G(i+N) = \bigcup_{j\in S'} \bot(j+P)$, where $S' = S$. For every $j \in S$, $\lceil j \rceil^G = i$ and, since $i \in \overline{\mathcal{L}}_{\bar{G}^1}$, then by definition of K, $j \in K$. Therefore by hypothesis, for every $j \in S$, $\lceil j+P \rceil^G = \lceil j \rceil^G + N = i + N$, hence $S \subseteq S'$. Moreover $\nexists j' \in \mathbb{Z} \setminus S$ s.t. $\lceil j'+P \rceil^G = i+N$. In fact if it existed, then, since Condition (1) is verified, then it would follow that $\lceil j' \rceil^G = i$ that is in contrast with $G(i) = \bigcup_{j\in S} \bot(j)$, since $j' \notin S$; hence $S \supseteq S'$ and we conclude $S = S'$.

(25) $\rightarrow$ (24). Suppose by contradiction that $\exists k \in \mathcal{L}_G$ s.t. (i) $(i+N) \notin \overline{\mathcal{L}}_G$ or (ii) if $G(k) = \bigcup_{j\in S} \bot(j)$ then, $G(k+N) \neq \bigcup_{j\in S} \bot(j+P)$. If $k \in \overline{\mathcal{L}}_{\bar{G}^1}$ then it is absurd by hypothesis; otherwise, from the periodicity of $\bar{G}^1$, follows that $\exists k' \in \overline{\mathcal{L}}_{\bar{G}^1}$ and $\exists \alpha \in \mathbb{Z}$ s.t. $G(k') = G(k + \alpha N_{\bar{G}^1}) = \bigcup_{j\in S} \bot(j + \alpha P_{\bar{G}^1})$. By hypothesis follows that $G(k'+N) \in \mathcal{L}_G$; then, from the periodicity of $\bar{G}^1$,

[5] The relation (24) $\leftrightarrow$ (25) can be easily proved too.

$G(k' + N - \alpha N_{\bar{G}^1}) \in \mathcal{L}_G$ and hence $(i + N) \in \overline{\mathcal{L}}_G$, in contradiction with (i). By hypothesis follows that $G(k' + N) = \bigcup_{j \in S} \perp(j + \alpha P_{\bar{G}^1} + P)$. Finally, from the periodicity of $\bar{G}^1$, follows that $G(k' + N - \alpha N_{\bar{G}^1}) = G(k + N) = \bigcup_{j \in S} \perp(j + P)$ in contrast with (ii).

B XML schemas

B.1 The *Calendar* XSD

```
<?xml version="1.0"?>
<xsd:schema xmlns="http://cs.uvm.edu/calendar"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://cs.uvm.edu/calendar"
 elementFormDefault="qualified">

<!-- Types -->
<xsd:simpleType name="identifier">
 <xsd:restriction base="xsd:token">
  <xsd:pattern value="([a-zA-Z][a-zA-Z0-9:\-\._]*)+"/>
 </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="expression">
 <xsd:restriction base="xsd:string">
  <xsd:pattern value="(\+|\-)?[0-9]+((\+|\-|\*|\/)?[0-9])*"/>
 </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="description">
 <xsd:simpleContent>
  <xsd:extension base="xsd:string"/>
 </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="granularityType">
 <xsd:sequence>
  <xsd:group ref="operationGroup"/>
 </xsd:sequence>
 <xsd:attribute name="defines" type="identifier" use="optional"/>
</xsd:complexType>

<!-- Root element -->
<xsd:element name="calendar">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element name="description" type="description"/>
   <xsd:any maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="primitive" type="identifier" use="optional"/>
 </xsd:complexType>
</xsd:element>

<!-- Operations -->
<xsd:element name="gran">
 <xsd:complexType>
  <xsd:attribute name="refid" type="identifier" use="required"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="equiv">
 <!-- equiv will not be supported in early implementations -->
 <xsd:complexType>
  <xsd:sequence>
```

```
   <xsd:group ref="operationGroup" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="group">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
  <xsd:attribute name="size" type="expression" use="required"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="alter">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0" maxOccurs="2"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
  <xsd:attribute name="offset" type="expression" use="required"/>
  <xsd:attribute name="repeat" type="expression" use="required"/>
  <xsd:attribute name="change" type="expression" use="required"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="shift">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
  <xsd:attribute name="offset" type="expression" use="required"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="subset">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
  <xsd:attribute name="start" type="expression" use="required"/>
  <xsd:attribute name="end" type="expression" use="required"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="select-down">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0" maxOccurs="2"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
  <xsd:attribute name="offset" type="expression" use="required"/>
  <xsd:attribute name="size" type="expression" use="required"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="select-up">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0" maxOccurs="2"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
 </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="select-by-overlap">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0" maxOccurs="2"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
  <xsd:attribute name="offset" type="expression" use="required"/>
  <xsd:attribute name="size" type="expression" use="required"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="union">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0" maxOccurs="2"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="intersect">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0" maxOccurs="2"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="difference">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:group ref="operationGroup" minOccurs="0" maxOccurs="2"/>
  </xsd:sequence>
  <xsd:attribute name="refid" type="identifier" use="optional"/>
 </xsd:complexType>
</xsd:element>

<xsd:group name="operationGroup">
 <xsd:choice>
  <xsd:element ref="gran"/>
  <xsd:element ref="equiv"/>
  <xsd:element ref="group"/>
  <xsd:element ref="alter"/>
  <xsd:element ref="shift"/>
  <xsd:element ref="subset"/>
  <xsd:element ref="select-down"/>
  <xsd:element ref="select-up"/>
  <xsd:element ref="select-by-overlap"/>
  <xsd:element ref="union"/>
  <xsd:element ref="intersect"/>
  <xsd:element ref="difference"/>
 </xsd:choice>
</xsd:group>

</xsd:schema>
```

B.2 The *Base* XSD example

```
<?xml version="1.0"?>
<xsd:schema xmlns="http://cs.uvm.edu/calendar/base"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:cal="http://cs.uvm.edu/calendar"
 targetNamespace="http://cs.uvm.edu/calendar/base"
 elementFormDefault="qualified">
```

```
<xsd:import namespace="http://cs.uvm.edu/calendar"
 schemaLocation="http://www.cs.uvm.edu/~jgustie/calendar.xsd"/>

<xsd:element name="second" type="cal:granularityType"/>
<xsd:element name="minute" type="cal:granularityType"/>
<xsd:element name="hour" type="cal:granularityType"/>
<xsd:element name="day" type="cal:granularityType"/>
<xsd:element name="week" type="cal:granularityType"/>
<xsd:element name="pseudo-month" type="cal:granularityType"/>
<xsd:element name="month" type="cal:granularityType"/>
<xsd:element name="year" type="cal:granularityType"/>
<xsd:element name="monday" type="cal:granularityType"/>
<xsd:element name="tuesday" type="cal:granularityType"/>
<xsd:element name="wednesday" type="cal:granularityType"/>
<xsd:element name="thursday" type="cal:granularityType"/>
<xsd:element name="friday" type="cal:granularityType"/>
<xsd:element name="saturday" type="cal:granularityType"/>
<xsd:element name="sunday" type="cal:granularityType"/>
<xsd:element name="weekday" type="cal:granularityType"/>
<xsd:element name="weekend" type="cal:granularityType"/>
<xsd:element name="moon-phase" type="cal:granularityType"/>
<xsd:element name="moon-period" type="cal:granularityType"/>
<xsd:element name="full-moon" type="cal:granularityType"/>
</xsd:schema>
```

B.3 The *Base* XML example

```
<?xml version="1.0"?>
<cal:calendar primitive="micro-second"
xmlns="http://cs.uvm.edu/calendar/base"
xmlns:cal="http://cs.uvm.edu/calendar">

 <cal:description>
  This is the base calendar.
 </cal:description>

 <second>
  <cal:group refid="micro-second" size="1000"/>
 </second>

 <minute>
  <cal:group refid="second" size="60"/>
 </minute>

 <hour>
  <cal:group refid="minute" size="60"/>
 </hour>

 <day>
  <cal:group refid="hour" size="24"/>
 </day>

 <week>
  <cal:group refid="day" size="7"/>
 </week>

 <pseudo-month defines="month">
  <!-- November -->
  <cal:alter refid="day" offset="11" repeat="12" change="-1">
   <!-- September -->
   <cal:alter refid="day" offset="9" repeat="12" change="-1">
    <!-- June -->
    <cal:alter refid="day" offset="6" repeat="12" change="-1">
     <!-- April -->
     <cal:alter refid="day" offset="4" repeat="12" change="-1">
      <!-- February -->
      <cal:alter refid="day" offset="2" repeat="12" change="-3">
```

```
      <!-- January, March, May, July, Augest, October, December -->
      <cal:group refid="day" size="31"/>
     </cal:alter>
    </cal:alter>
   </cal:alter>
  </cal:alter>
 </cal:alter>
</pseudo-month>

<month>
 <!--4790=2+12*399 and 4800=12*400-->
 <cal:alter refid="day" offset="4790" repeat="4800" change="1">
  <!--1190=2+12*99 and 1200=12*100-->
  <cal:alter refid="day" offset="1190" repeat="1200" change="-1">
   <!--38=2+12*3 and 48=12*4-->
   <cal:alter refid="day_pseudo-month" offset="38" repeat="48" change="1"/>
  </cal:alter>
 </cal:alter>
</month>

<year>
 <cal:group refid="month" size="12"/>
</year>

<monday>
 <cal:select-down refid="day_week" offset="1" size="1"/>
</monday>

<tuesday>
 <cal:select-down refid="day_week" offset="2" size="1"/>
</tuesday>

<wednesday>
 <cal:select-down refid="day_week" offset="3" size="1"/>
</wednesday>

<thursday>
 <cal:select-down refid="day_week" offset="4" size="1"/>
</thursday>

<friday>
 <cal:select-down refid="day_week" offset="5" size="1"/>
</friday>

<saturday>
 <cal:select-down refid="day_week" offset="6" size="1"/>
</saturday>

<sunday>
 <cal:select-down refid="day_week" offset="7" size="1"/>
</sunday>

<weekday>
 <cal:select-down refid="day_week" offset="2" size="5"/>
</weekday>

<weekend>
 <cal:union refid="saturday_sunday"/>
</weekend>

<moon-phase>
 <cal:group size="25514429">
  <cal:shift refid="micro-second" offset="-62947536781000"/>
 </cal:group>
</moon-phase>

<moon-period>
 <cal:group refid="moon-phase" size="100"/>
```

```
 </moon-period>

 <full-moon>
  <cal:select-down refid="moon-phase_moon-period" offset="50" size="2"/>
 </full-moon>

</cal:calendar>
```

B.4 The *Output* XSD

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd"
   xmlns="http://tempuri.org/XMLSchema.xsd"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified">

 <!--types-->
 <xs:complexType name="intervalType">
  <xs:attribute name="from" type="xs:unsignedLong" use="required" />
  <xs:attribute name="to" type="xs:unsignedLong" use="required" />
 </xs:complexType>

 <xs:complexType name="granulType">
  <xs:sequence maxOccurs="unbounded">
   <xs:element name="interval" type="intervalType" />
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="granularityType">
  <xs:sequence maxOccurs="unbounded">
   <xs:element name="granul" type="granulType" />
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" />
  <xs:attribute name="period" type="xs:unsignedLong" />
  <xs:attribute name="lowerBound" type="xs:long" use="optional" />
  <xs:attribute name="upperBound" type="xs:long" use="optional" />
 </xs:complexType>

 <!--root-->
 <xs:element name="calendar">
  <xs:complexType>
   <xs:sequence maxOccurs="unbounded">
    <xs:element name="granularity" type="granularityType" />
   </xs:sequence>
   <xs:attribute name="name" type="xs:string" use="required" />
   <xs:attribute name="namespace" type="xs:string" use="required" />
   <xs:attribute name="bottomGranularity" type="xs:string" use="required" />
  </xs:complexType>
 </xs:element>
</xs:schema>
```

B.5 The *Output* XML example

```
<?xml version="1.0" ?>
<calendar xmlns="http://tempuri.org/XMLSchema.xsd"
   name="test" namespace="http://example" bottomGranularity="hour">

 <granularity name="day" period="24">
  <granul>
   <interval from="1" to="24"/>
  </granul>
 </granularity>

 <granularity name="weekendday" period="168">
  <granul>
   <interval from="120" to="144"/>
```

```
  </granul>
  <granul>
   <interval from="144" to="168"/>
  </granul>
 </granularity>

 <granularity name="working-hour" period="24">
  <granul>
   <interval from="8" to="12"/>
   <interval from="14" to="18"/>
  </granul>
 </granularity>
</calendar>
```

C Notion

In the text the following symbols were used:

- G, G_1, G_2 and H indicate granularities;
- $\perp$ indicates the bottom granularity;
- $\trianglelefteq$ indicates the *groups* relation between granularities;
- $\preceq$ indicates the *finer then* relation between granularities;
- $\bar{\trianglelefteq}$ indicates the *groups periodically into* relation between granularities;
- P_G, N_G and R_G indicate respectively the period, the period label distance and the number of granules in a period of a granularity G;
- $\lceil n \rceil_H^G$ and $\lfloor n \rfloor_H^G$ represent the *up* and *down* operation respectively;
- $\lceil n \rceil^G$ and $\lfloor n \rfloor^G$ represent the *up* and *down* operation respectively when the second operator is the bottom granularity;
- $\mathcal{L}_G$ represents the label set of a granularity G;
- $\mathcal{L}_G^P$ represents the label set of a granularity G in one of its periods;
- $l_G = \lceil i \rceil^G$ where i is the smallest positive integer such that $\lceil i \rceil^G$ is defined;
- $\overline{\mathcal{L}}_G = \{j \in \mathcal{L}_G | l \leq j < l + N_G\}$;
- $\hat{\mathcal{L}}_G$ is the set of labels of granules of G that cover one in $\perp(1) \ldots \perp(P_G)$;
- $\hat{\mathcal{L}}_G^{P_{G'}}$ is the set of labels of granules of G that cover one in $\perp(1) \ldots \perp(P_{G'})$.