2009 16th International Symposium on Temporal Representation and Reasoning

Building Logical Specifications of Temporal Granularities through Algebraic Operators

Carlo Combi and Sara Degani

Department of Computer Science, University of Verona (Italy)
Strada le Grazie 15, I-37134 Verona, Italy
e-mail: {*carlo.combi, sara.degani*}*@univr.it*

***Abstract*—Logical and algebraic approaches have been proposed in the literature for the definition and the management of time granularities. In the algebraic framework, a bottom granularity is assumed, and new granularities are created by means of suitable algebraic operators. In the logical framework, mathematical structures are used to represent time granularities, and logical formulae are used to specify meaningful properties betweeen them. In this paper we extend the logical approach of Combi *et al.* [3] for representing and reasoning about temporal granularities, defining logical formulae corresponding to calendar algebra operators. This approach allows one to define a large set of granularities, possibly corresponding to complex formulae, in a natural way. Indeed, it gives the possibility of specifying granularity-related formulae in an easier way through operators that are intuitive and natural as the algebraic ones, still maintaining the capability of reasoning on granularities through the (formal) techniques that have been deeply investigated for the logical framework.**

***Keywords*-temporal granularity; temporal logics; granularity specifications.**

I. INTRODUCTION

A fact of the real world is usually described with reference to the temporal dimension, expressed by means of time units such as day, month, year. A time unit can be represented by means of a *time granularity*, i.e., the representation on a temporal domain of groups of elements that are perceived as inseparable units called *granules*. Furthermore, in several research areas of computer science, such as formal specifications, data mining, and temporal databases it is considered of primary importance the capability of providing temporal representations of the temporal dimension of a fact at different levels of granularity, and of managing the relationships between them.

Logical and algebraic approaches have been proposed for the definition and the management of time granularities in the literature [3], [4], [7]. In the logical approach, mathematical structures are used to represent granularities and relationships between them; these structures consist of a possibily infinite set of different time domains. Specific operators allows one either to express properties in a given time domain or to relate different time domains. Properties involving different time granularities are then expressed by means of logical formulae defined over these structures. The algebraic approach assumes a bottom granularity, and defines a finite number of algebraic operators that generate new granularities from existing ones. A time granularity may therefore be identified by means of an algebraic expression [7].

Algebraic operators are intuitive and new granularities may be easily derived from existing ones. Nevertheless, in algebraic frameworks few attention is given to the investigation of algorithms checking the existence of meaningful relationships between granularities, e.g., verifying if the granularity G_1 is equivalent to the granularity G_2; moreover, only a finite number of time granularities may be expressed in this kind of approach through a finite number of expressions.

In this paper we consider the logical framework of Combi *et al.* [3], that represents time granularities by means of *labelled linear time structures*, identified by suitable Past Propositional Linear Time Logic formulae; this framework allows one to express both anchored and unanchored granularities (i.e., possibly infinite granule sets, anchored to different time points on the time domain). However, formulae corresponding to real world granularities may correspond to long formulae, not easy to read and understand. The goal of the approach we present here is to allow one to define a large set of granularities, possibly corresponding to complex formulae, in a natural way, as it happens with the algebraic approach. To this end, we show how the basic operators of the calendar algebra proposed by Ning *et al.* [7] can be expressed on a labelled linear time structure by means of suitable logical PPLTL formulae. This way, it is possible to specify granularity-related formulae in an easier way through operators that are intuitive and natural as the algebraic ones, still maintaining the capability of reasoning on granularities through the (formal) techniques that have been deeply investigated for the logical framework [3].

The paper is structured as follows. Section II introduces time granularities and presents the considered logical and algebraic frameworks; Section III defines new logical formulae encoding algebraic operators. Section IV concludes the paper with final discussion and remarks.

978-0-7695-3727-6/09 $25.00 © 2009 IEEE
DOI 10.1109/TIME.2009.19

II. Background

The main approaches for representing and managing time granularities, we will consider in this paper, are the *logical* [3], [4] and the *algebraic* [7] ones. Before presenting in sections II-A and II-B the logical framework of Combi *et al.* [3], and the algebraic approach of Ning *et al.* [7], let us start with the introduction of some basic and widely acknowledged notions about time granularities [1].

Definition 1: A *time domain* is a pair $(T, \leq)$, where T is a non-empty set of *time instants* and $\leq$ is a total order on T. $\square$

A time domain is a set of primitive temporal entities used to define and interpret time-related concepts, and ordered according to the relationship $\leq$. Examples of time domains are natural numbers $(\mathbb{N}, \leq)$, integers $(\mathbb{Z}, \leq)$, rational $(\mathbb{Q}, \leq)$ and real numbers $(\mathbb{R}, \leq)$.

Definition 2: A *granularity* is a mapping $G : \mathbb{Z} \rightarrow 2^T$ such that (1) if $i < j$ and $G(i)$ and $G(j)$ are non-empty, then each element of $G(i)$ is less than all the elements of $G(j)$, and (2) if $i < k < j$ and $G(i)$ and $G(j)$ are non-empty, then $G(k)$ is non-empty. $\square$

The domain of a granularity G is called *index set*, and an element of the codomain $G(i)$ is called a *granule*.

A granularity G is said *externally continuous* if there are no gaps between non-empty granules of G; *internally continuous* if there are no gaps inside the granules of G; *continuous* if it is both externally and internally continuous.

Given two granularities G_1 and G_2, a number of meaningful relationships can be estabilished between them:

- $\mathbf{FinerThan}(G_1, G_2)$ holds if for each index i, there exists an index j such that $G_1(i) \subseteq G_2(j)$. For instance, $\mathbf{FinerThan}(\mathsf{day}, \mathsf{month})$ holds.
- $\mathbf{SubGranularity}(G_1, G_2)$ holds if for each index i there exists an index j such that $G_1(i) = G_2(j)$. For example, $\mathbf{Subgranularity}(\mathsf{Sunday}, \mathsf{day})$ holds.
- $\mathbf{Group}(G_1, G_2)$ holds if for each index i, there exists a set S (either empty or in the form $\{j, j+1, \ldots, j+k\}$) such that $G_2(i) = \bigcup_{l \in S} G_1(l)$. For example, $\mathbf{Group}(\mathsf{day}, \mathsf{week})$ holds.
- $\mathbf{Shift}(G_1, G_2)$ holds if there exists $p \geq 0$ such that, for each index i, $G_2(i) = G_1(i + p)$.
- $\mathbf{Partition}(G_1, G_2)$ holds if both $\mathbf{Group}(G_1, G_2)$ and $\mathbf{FinerThan}(G_1, G_2)$ hold.

A. The Logical Framework

In the logical framework introduced in [3], time granularities are modelled by means of a specialized version of Definition 2, assuming that both the index set and the domain of granules are the linear discrete domain $(\mathbb{N}, \leq)$. Moreover, an internally continuous time granularity G is represented by means of a *labelled linear time structure*, identified by a suitable linear time formula in the context of Past Propositional Linear Time Logic (PPLTL). More specifically, a linear temporal axis is considered, and the granules of a granularity G are represented on it by means of the propositional symbols P_G and Q_G: the starting point of a granule of G in the structure is labelled by P_G, the ending point is labelled by Q_G.

Definition 3: Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ be a finite set of granularites, (a *calendar*), and $\mathcal{P}_\mathcal{G} = \{P_{G_i}, Q_{G_i} | 1 \leq i \leq n\}$ a finite set of proposition symbols associated with the calendar $\mathcal{G}$. Given an alphabet of proposition symbols $\mathcal{P} \supseteq \mathcal{P}_\mathcal{G}$, a $\mathcal{P}$-labelled linear time structure has the form $(\mathbb{N}, <, V)$, where $(\mathbb{N}, <)$ is the set of natural numbers with the usual order and $V : \mathbb{N} \rightarrow 2^\mathcal{P}$ is a labelling function that maps natural numbers to sets of proposition symbols. $\square$

Not every assignment of proposition symbols to the timeline by means of the labelling function V defines a time granularity. Therefore, the framework introduces the notion of *consistency* of a labelled structure with respect to a temporal granularity G, allowing one to identify assignments that define time granularities: a labelled linear time structure $(\mathbb{N}, <, V)$ is said *G-consistent* if every point labelled with P_G matches with a unique point labelled with Q_G, and vice versa.

The syntax of PPLTL is defined as follows.

Definition 4 (Syntax of PPLTL): Formulae of PPLTL are inductively defined as follows:

- a proposition symbol $P \in \mathcal{P}$ is a PPLTL formula;
- if ϕ and ψ are PPLTL formulae, then $\phi \wedge \psi$ and $\neg\phi$ are PPLTL formulas;
- if ϕ and ψ are PPLTL formulae, then $\mathbf{X}\phi$, $\phi\mathbf{U}\psi$, $\mathbf{X}^{-1}\phi$ and $\phi\mathbf{S}\psi$ are PPLTL formulae.

$\square$

Moreover:

- $\phi \vee \psi$ is defined as: $\neg(\neg\phi \wedge \neg\psi)$;
- $\phi \rightarrow \psi$ is defined as: $\neg\phi \vee \psi$;
- $\phi \leftrightarrow \psi$ is defined as: $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$;
- $\mathbf{F}\alpha$ (α will hold in the future) is defined as: $\mathtt{true}\mathbf{U}\alpha$;
- $\mathbf{G}\alpha$ (α will always hold in the future) is defined as: $\neg\mathbf{F}\neg\alpha$;
- $\mathbf{P}\alpha$ (α held in the past) is defined as: $\mathtt{true}\mathbf{S}\alpha$;
- $\mathbf{H}\alpha$ (α always held in the past) is defined as: $\neg\mathbf{P}\neg\alpha$.

where $\mathtt{true} = P \vee \neg P$, for a some $P \in \mathcal{P}$.
Temporal operators in $\{\mathbf{X}, \mathbf{U}, \mathbf{X^{-1}}, \mathbf{S}\}$ have priority over boolean operators $\{\wedge, \vee\}$; moreover $\neg$ has priority over $\wedge$ and over $\vee$, and $\wedge$ has priority over $\vee$.

PPLTL logic is interpreted in $\mathcal{P}$-labelled linear time structures. The semantics of PPLTL is defined as follows.

Definition 5 (Semantics of PPLTL): Let $\mathcal{M} = (\mathbb{N}, <, V)$ be a $\mathcal{P}$-labelled linear time structure and $i \in \mathbb{N}$. The truth of a PPLTL formula ϕ in $\mathcal{M}$ with respect to the point i, denoted with $\mathcal{M}, i \models \phi$, is defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, i \models P & \text{iff} & P \in V(i) \text{ for } P \in \mathcal{P}; \\
\mathcal{M}, i \models \phi \wedge \psi & \text{iff} & \mathcal{M}, i \models \phi \text{ and } \mathcal{M}, i \models \psi; \\
\mathcal{M}, i \models \neg\phi & \text{iff} & \text{is not the case that } \mathcal{M}, i \models \phi; \\
\mathcal{M}, i \models \phi\mathbf{U}\psi & \text{iff} & \mathcal{M}, j \models \psi \text{ for some } j \geq i \text{ and} \\
 & & \mathcal{M}, k \models \phi \text{ for each } i \leq k < j; \\
\mathcal{M}, i \models \mathbf{X}\phi & \text{iff} & \mathcal{M}, i+1 \models \phi; \\
\mathcal{M}, i \models \phi\mathbf{S}\psi & \text{iff} & \mathcal{M}, j \models \psi \text{ for some } j \leq i \text{ and} \\
 & & \mathcal{M}, k \models \phi \text{ for each } j < k \leq i; \\
\mathcal{M}, i \models \mathbf{X^{-1}}\phi & \text{iff} & i > 0 \text{ and } \mathcal{M}, i-1 \models \phi.
\end{array}
$$

□

We say that:

- $\mathcal{M}$ *is a model of* ϕ if $\mathcal{M}, 0 \models \phi$;
- a formula ϕ is *satisfiable* if and only if there exists a $\mathcal{P}$-labelled linear time structure that is a model of ϕ;
- a formula ϕ is *valid* if and only if every $\mathcal{P}$-labelled linear time structure is a model ϕ.

Note that ϕ is valid if and only if $\neg\phi$ is not satisfiable.

A PPLTL formula intensionally defines a possibly infinite set of labelled linear time structures (the set of the models of the formula). Since a linear time structure defines a granularity, linear time logic formulae can be used to define possibly infinite sets of granularities. In the following, according to [3], we show some examples of PPLTL formulae defining sets of time granularities.

The set of internally continuous granularities is captured by the following PPLTL formula:

$$\mathbf{IntContGran}(G) = \mathbf{G}((P_G \rightarrow \alpha) \wedge (Q_G \rightarrow \beta)),$$

where

$$
\begin{array}{lll}
\alpha & \text{is} & Q_G \vee \mathbf{X}(\neg(P_G \vee Q_G)\mathbf{U}(\neg P_G \wedge Q_G)), \text{ and} \\
\beta & \text{is} & P_G \vee \mathbf{X^{-1}}(\neg(P_G \vee Q_G)\mathbf{S}(P_G \wedge \neg Q_G)).
\end{array}
$$

The first conjunct ($P_G \rightarrow \alpha$) of the formula states that every point labelled with P_G matches with a unique point labelled with Q_G, while the second conjunct ($Q_G \rightarrow \beta$) states that every point labelled with Q_G matches with a unique point labelled with P_G; in other words, they capture the definition of consistency of a granularity. The set of continuous granularities is captured by the formula $\mathbf{ContGran}(G)$, defined in a similar way [3].

Binary relations between granularities may be expressed by means of PPLTL formulae involving a finite number of different granularities, represented with different pairs of marking proposition symbols.

The relationship $\mathbf{FinerThan}(G_1, G_2)$ is captured by the following formula:

$$
\begin{array}{l}
\mathbf{IntContGran}(G_1) \wedge \mathbf{IntContGran}(G_2) \wedge \\
\mathbf{G}(P_{G_1} \rightarrow ((\neg P_{G_2} \wedge \neg Q_{G_2})\mathbf{S}P_{G_2} \wedge \\
(\neg P_{G_2} \wedge \neg Q_{G_2})\mathbf{U}Q_{G_1})).
\end{array}
$$

The formula requires each granule of G_1 to be included into some granule of G_2; more than one granule of G_1 may be included in the same granule of G_2.

The relationship $\mathbf{SubGranularity}(G_1, G_2)$ is expressed in a similar way [3].

The above proposal copes with internally continuous granularities. In [3] it is extended to also cope with granularities with gaps inside the granules. The set of proposition symbols associated with the granularity G is extended to include the symbols P_{H_G} and Q_{H_G}, that are used to delimit any gap inside the granules of G. The description of gaps of G is therefore the description of a granularity H_G, such that $\mathbf{FinerThan}(H_G, G)$ holds. The definition of consistency is extended to also consider gaps inside the granules of a granularity: a labelled linear time structure is said *G-gap-consistent* if it is G-consistent and H_G-consistent, if each granule of H_G is a subset of some granule of G, and if no granule of G is the union of some granules of H_G.

A G-gap-consistent linear time structure corresponds to a (not necessarely continuous) granularity, and vice versa. The set of (not necessarely continuous) granularities $\mathbf{Gran}(G)$ is then expressed by the following formula:

$$\mathbf{FinerThan}(H_G, G) \wedge \mathbf{G}(P_G \rightarrow \neg\alpha)$$

where α is

$$
\begin{array}{c}
P_{H_G} \wedge ((Q_G \wedge Q_{H_G}) \vee \\
\mathbf{X}((\neg Q_G \wedge (Q_{H_G} \rightarrow \mathbf{X}P_{H_G}))\mathbf{U}(Q_G \wedge Q_{H_G}))).
\end{array}
$$

B. The Algebraic Framework

The algebraic framework presented in [7] considers an extension of Definition 2, that associates a label to each granule. A labelled granularity is a pair ($\mathcal{L}$, G), where $\mathcal{L}$ is a (possibly non contiguous) subset of $\mathbb{Z}$, and G is a mapping from $\mathcal{L}$ to the subset of the time domain T; $\mathcal{L}$ is the set of labels of G, and its elements may be used to identify the granules of a granularity.

An extension of some of the relationships between granularities to labelled granularities is then provided [7]. In particular:

- $\mathbf{LabelledAlignedSubgranularity}(G_1, G_2)$ holds if each label i of G_1 is also a label of G_2, and $G_1(i) = G_2(i)$. For instance, if each Sunday has the same label of the corresponding day, then $\mathbf{LabelledAlignedSubgranularity}(\mathsf{Sunday}, \mathsf{day})$ holds.
- $\mathbf{Partition}(G_1, G_2)$ hols if both $\mathbf{Group}(G_1, G_2)$ and $\mathbf{FinerThan}(G_1, G_2)$ hold. For instance, $\mathbf{Partition}(\mathsf{day}, \mathsf{week})$ holds.

The algebraic approach defines a symbolic representation of granularities termed *calendar algebra* [7]. This representation captures the relationships between the granularities of a calendar, and it is composed of two kinds of operations: *grouping-oriented* operations and *granule-oriented* operations. Grouping oriented operations combine certain granules of a granularity together to form the granules of the new granularity, while granule oriented operations make choices of which granules should remain in the new granularity. In the following we present the grouping oriented operations, and the granule-oriented operation *Select-down*.

1) The Grouping Operation: The grouping operation is defined as follows [7].

Definition 6: Let G be a granularity, and m a positive. The grouping operation $\mathbf{Group_m}(G)$ generates a new granularity G' by partitioning the granules of G in groups of m granules; each group is a granule of G'. Formally, for each integer i, $G'(i) = \bigcup_{j=(i-1)\cdot m+1}^{i\cdot m} G(j)$. $\square$

For example, $\mathsf{week} = \mathbf{Group_7}(\mathsf{day})$.

2) The Altering-tick Operation: Let G_1 and G_2 be two granularities, such that $\mathbf{Partition}(G_2, G_1)$ holds, and let l, k, m be integers such that $1 \leq l \leq m$. Since $\mathbf{Partition}(G_2, G_1)$ holds, each granule of G_1 is composed of some contiguous granules of G_2; moreover, the granules of G_1 may be partitioned in groups of m granules. The altering-tick operation $\mathbf{Alter^m_{l,k}}(G_2, G_1)$ modifies the granules of G_1 so that the lth granule of each of each group has $|k|$ additional (or fewer when $k < 0$) granules of G_2. For each $i = l + m \cdot n$, with $n \in \mathbb{Z}$, $G_1(i)$ is the granule that is expanded (or shrunk). If $i > 0$, $G_1(i)$ 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; if $i < 0$, $G_1(i)$ 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.

Definition 7: Let G_1 and G_2 be two granularities, such that $\mathbf{Partition}(G_2, G_1)$ holds, and let l, k, m be integers such that $1 \leq l \leq m$. For each integer i such that $G_2(i) \neq \emptyset$, let b_i and t_i two integers such that $G_2(i) = \bigcup_{j=b_i}^{t_i} G_1(j)$ (they exist because $\mathbf{Partition}(G_1, G_2)$ holds). Then $G' = \mathbf{Alter^m_{l,k}}(G_1, G_2)$ is the granularity such that for each integer i:

$$G'(i) = \begin{cases} \emptyset & \text{if} \quad G_2(i) = \emptyset \\ \bigcup_{j=b'_i}^{t'_1} G_1(j) & \text{otherwise} \end{cases}$$

where

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

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

and

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

$\square$

For example, if $G = \mathbf{Group_{30}}(\mathsf{day})$ (i.e., G has granules of 30 days), the operation $\mathbf{Alter^{12}_{5,1}}(\mathsf{day}, G)$ adds a day ($k = 1$) to the fifth group of each set of 12 of such groups ($l = 5, m = 12$).

3) The Shifting Operation: The shifting operation is defined as follows [7].

Definition 8: Let G be a granularity, and let m be an integer. The operation $\mathbf{Shift_m}(G)$ generates a new granularity G' by performing a shift of m positions of the labels of G. Formally, for each integer i, $G'(i) = G(i-m)$. $\square$

The shifting operation can be used to express time differences. For example, since the hours of New York are three hours later than those of Los Angeles, the hours of New York can be generated by $\mathsf{LA\text{-}Hour} = \mathbf{Shift_{-3}}(\mathsf{NY\text{-}Hour})$, where the granularities NY-Hour and LA-Hour stand for the hours of New York and Los Angeles respectively.

4) The Combining Operation: The combining operation is defined as follows [7].

Definition 9: Let G_1 and G_2 be two granularities with label sets $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. The operation $\mathbf{Combine}(G_1, G_2)$ generates a new granularity G' by combining in a granule of G' all the granules of G_2 that are included in a granule of G_1. Formally, for each $i \in \mathcal{L}_1$, let

$$s(i) = \begin{cases} \emptyset & \text{if} \quad G_1(i) = \emptyset, \\ \{j \in \mathcal{L}_2 | \emptyset \neq G_2(j) \subseteq G_1(i)\} & \text{otherwise.} \end{cases}$$

Then, $G' = \mathbf{Combine}(G_1, G_2)$ is the granularity with label set $\mathcal{L}_{G'} = \{i \in \mathcal{L}_1 | s(i) \neq \emptyset\}$ such that for each $i \in \mathcal{L}_{G'}$, $G'(i) = \bigcup_{j \in s(i)} G_2(j)$. $\square$

For example, given the granularities month and businessDay, the granularity for business months can be generated by $\mathsf{businessMonth} = \mathbf{Combine}(\mathsf{month}, \mathsf{businessDay})$, since each business month is the union of all the business days in the month.

5) The Anchored-grouping Operation: The anchored-grouping operation is defined as follows [7].

Definition 10: Let G_1 be a simple granularity and G_2 be a granularity with label set $\mathcal{L}_2$, such that $\mathbf{LabelAlignedSubGranularity}(G_2, G_1)$ holds. The operation $\mathbf{Anchored\text{-}group}(G_1, G_2)$ generates a new granularity G' by combining in a granule of G' all the granules of G_1 that are between two granules of G_2. Formally, $G' = \mathbf{Anchored\text{-}group}(G_1, G_2)$ is the granularity with label set $\mathcal{L}_{G'} = \mathcal{L}_2$ such that for each $i \in \mathcal{L}_{G'}$, $G'(i) = \bigcup_{j=1}^{i'-1} G_1(j)$, where i' is the next label of G_2 after i. $\square$

For example, if a fiscal year begins in October and ends in the next September, the granularity that corresponds to fiscal years can be generated by $\mathsf{fiscal\text{-}year} = \mathbf{Anchored\text{-}group}(\mathsf{month}, \mathsf{October})$.

6) The Select-down operator: The select-down operation is defined as follows [7].

Definition 11: Let G_1 and G_2 be two granularities and let k, l two integers such that $k \neq 0$ and $l > 0$ be two integers. The operation $\mathbf{Select\text{-}down}^l_k(G_1, G_2)$ selects granules of G_1 by picking up l granules starting from the k_{th} one in each set of granules of G_1 that are contained in a granule of G_1.

Formally, $G' = \mathbf{Select\text{-}down}_k^l(G_1, G_2)$ is the granularity with label set $\mathcal{L}_{G'} = \bigcup_{i\in\mathcal{L}_2} \delta_k^l(\{j \in \mathcal{L}_1 | \emptyset \neq G_1(j) \subseteq G_2(i)\})$, and for each $i \in \mathcal{L}_{G'}$, $G'(i) = G_1(i)$. $\Box$

For example, $\mathsf{Sunday} = \mathit{Select\text{-}down}_1^1(\mathsf{day}, \mathsf{week})$

III. Logical Definition of Algebraic Operators

In this section we define the previously described algebraic operators by means of logical formulae according to the framework presented in Section II-A. In the following, instead of providing formal proofs that the introduced formulae are equivalent to the corresponding algebraic expressions, we discuss and introduce the meaning of the different parts of formulae, to allow one to have an overall comprehension of the proposed approach. Examples throughout this section and a real-world example related to the medical domain at the end of this section will complete the description of the introduced formula.

A. The Grouping Operation

We define the grouping operation $Group(G_1, G_2, m)$ as follows.

Definition 12: Let G_1 be a granularity, and m an integer. The granularity G_2 obtained through the algebraic operator $\mathbf{Group_m}(G_1)$, may be specified by the following PPLTL formula:

$$Group(G_1, G_2, m) =$$
$$\mathbf{Gran}(G_1) \wedge \mathbf{Gran}(G_2) \wedge (\neg P_{G_1} \wedge \neg P_{G_2})\mathbf{U}(P_{G_1} \wedge P_{G_2}) \wedge$$
$$\mathbf{G}(P_{H_{G_2}} \rightarrow (P_{H_{G_1}} \vee (\neg P_{G_1} \wedge \mathbf{X^{-1}} Q_{G_1})) \wedge$$
$$Q_{H_{G_2}} \rightarrow (Q_{H_{G_1}} \vee (\neg Q_{G_1} \wedge \mathbf{X} P_{G_1}))) \wedge$$
$$\mathbf{G}(P_{G_2} \rightarrow \alpha_{m-1}),$$

where α_0 is

$$P_{G_1} \rightarrow \neg(Q_{G_1} \vee Q_{G_2})\mathbf{U}(Q_{G_1} \wedge Q_{G_2}) \wedge$$
$$\neg P_{G_1} \rightarrow (\neg P_{G_1})\mathbf{U}(P_{G_1} \wedge \neg(Q_{G_1} \vee Q_{G_2})\mathbf{U}(Q_{G_1} \wedge Q_{G_2})),$$

and, for $k > 0$, α_k is recursively defined as

$$P_{G_1} \rightarrow \neg(Q_{G_1} \vee Q_{G_2})\mathbf{U}(Q_{G_1} \wedge \neg Q_{G_2} \wedge \mathbf{X}_{\alpha_{\mathbf{k-1}}}) \wedge \neg P_{G_1} \rightarrow (\neg P_{G_1})\mathbf{U}(P_{G_1} \wedge \neg(Q_{G_1} \vee Q_{G_2})\mathbf{U}(Q_{G_1} \wedge \neg Q_{G_2} \wedge \mathbf{X}_{\alpha_{\mathbf{k-1}}})).$$

$\Box$

With the first two conjuncts, the formula requires G_1 and G_2 to be (possibly non continuous) granularities. The third conjunct requires G_1 and G_2 to begin at the same time instant; the fourth conjunct states that a gap in G_2 may exists only if there exists a gap in a granule of G_1, or if there exists a gap between two granules of G_1 that belong to the same granule of G_2. The last conjunct states that a granule of G_2 is composed of m consecutive granules of G_1. Figure 1 shows an example where $m = 2$.

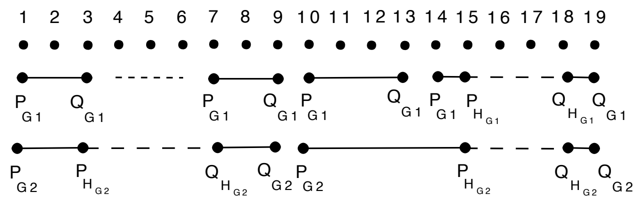

Figure 1. The grouping operation

B. The Altering-tick Operation

In order to define a logical formula corresponding to the operator $\mathbf{Alter^m_{l,k}}(G_1, G_2)$, we assume l, k, m to be positive integers (i.e., we only consider the expansion of the granules of G_2 with later granules of G_1). Moreover, we need to make a further assumption. The granules of G_3 are defined by expanding some granules of G_2. In order to estabilish the conditions on the beginning and the end of the granules of G_3 it is necessary to quantify such expansion, since it affects the definition of later granules of G_3. Figure 2 represents the grouping of the granules of G_2 in groups of two granules (m=2); for the first two of such groups (p=2) the operation expands the second granule (l=2) with a granule of G_1 (k=1). In order to mark the beginning and the end of the granule of G_3 corresponding to the second group of granules of G_2, it is necessary to consider the expansion of the earlier group; indeed, the label P_{G_3} of the second granule of G_3 is shifted of a granule of G_1 with respect to the third granule of G_2. More in general, the size of the expansion depends on which group of m granules of G_2 we consider to generate a granule of G_3. Since the number of such groups cannot be a priori determined, it is not possible to define a formula that captures the altering-thick operation on all of these groups; we can however define a formula capturing the operation on the first p groups, with $p > 0$. We therefore define a new operation $Alter^{m,p}_{l,k}(G_1, G_2, G_3)$ that performs the altering thick operation on the first p groups of m granules of G_2.

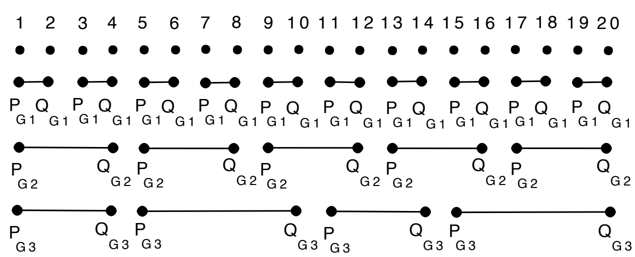

Figure 2. The altering-tick operation

Definition 13: Let G_1 and G_2 be two granularities such that $\mathbf{Partition}(G_1, G_2)$ holds, and let l, k, m, p be positive integers such that $1 \leq l \leq m$. The granularity G_3 obtained through the algebraic operator $\mathbf{Alter^m_{l,k}}(G_2, G_1)$ (for the first p groups of m granules of G_2), may be specified by the following PPLTL formula:

$$Alter^{m,p}_{l,k}(G_1, G_2, G_3) =$$
$$\mathbf{ContGran}(G_1) \wedge \mathbf{ContGran}(G_2) \wedge \mathbf{ContGran}(G_3) \wedge$$
$$\mathbf{Partition}(G_1, G_2) \wedge \bigwedge_{i=0}^{p} \alpha_{m\cdot i}$$

where, for $n > 0$,

$$\begin{aligned}
\alpha_0 =& P_{G_2} \rightarrow \delta_{l-1} \\
\alpha_n =& (\neg Q_{G2})\mathbf{U}(Q_{G_2} \wedge \mathbf{X}\alpha_{n-1}) \\
\delta_0 =& \beta_{k\cdot i} \wedge (\neg Q_{G_2})\mathbf{U}(Q_{G_2} \wedge \gamma_{k\cdot(i+1)})\wedge \\
& \mathbf{X}((\neg P_{G_2})\mathbf{U}(P_{G_2} \rightarrow \phi_{m-l-1})) \\
\delta_n =& \beta_{k\cdot i} \wedge (\neg Q_{G_2})\mathbf{U}(Q_{G_2} \wedge \gamma_{k\cdot i})\wedge \\
& \mathbf{X}((\neg P_{G_2})\mathbf{U}(P_{G_2} \rightarrow \delta_{n-1})) \\
\phi_0 =& \beta_{k\cdot(i+1)} \wedge (\neg Q_{G_2})\mathbf{U}(Q_{G_2} \wedge \gamma_{k\cdot(i+1)}) \\
\phi_n =& \beta_{k\cdot(i+1)} \wedge (\neg Q_{G_2})\mathbf{U}(Q_{G_2} \wedge \gamma_{k\cdot(i+1)})\wedge \\
& \mathbf{X}((\neg P_{G_2})\mathbf{U}(P_{G_2} \rightarrow \phi_{n-1})) \\
\beta_0 =& P_{G_3} \\
\beta_n =& (\neg Q_{G_1})\mathbf{U}(Q_{G_1} \wedge \mathbf{X}\beta_{n-1}) \\
\gamma_0 =& Q_{G_3} \\
\gamma_n =& (\neg Q_{G_1})\mathbf{U}(Q_{G_1} \wedge \mathbf{X}\gamma_{n-1})
\end{aligned}$$

□

The first three conjuncts require G_1, G_2 and G_3 to be continuous granularities; the formula $\mathbf{Partition}(G_1, G_2)$ is defined as $\mathbf{Group}(G_1, G_2) \wedge \mathbf{FinerThan}(G_1, G_2)$. The formula $\alpha_{m\cdot i}$ identifies the ith group of m granules of G_2; in such group, the formula δ_n, for $n > 0$, estabilishes that the granules of G_3 corresponding to the granules of G_2 before the lth granule, are shifted of $k \cdot i$ granules of G_1 with respect to the corresponding granules of G_2. The formula δ_0 estabilishes that the granule of G_3 corresponding to the lth granule of G_2 of the considered group, begins $k \cdot i$ granules of G_1 after the beginning of the corresponding granule of G_2, and ends $k \cdot (i + 1)$ granules of G_1 after the ending of the same granule. Finally, the formula ϕ_n, with $n \geq 0$, requires the granules of G_3 corresponding to the granules of G_2 after the lth granule in the considered group, to be shifted of $k \cdot (i + 1)$ granules of G_1 with respect to the granules of G_2.

Figure 2 shows the granularities identified by the formula $Alter^{2,2}_{2,1}(G_1, G_2, G_3)$. The formula α_2 identifies the second group of two granules of G_2; δ_1 estabilishes that the granule of G_3 corresponding to the first granule of G_2 in such group, is shifted of 1 granule of G_1 with respect to the same granule. Such shifting is caused by the expansion performed on the (only) earlier group of 2 granules of G_2. The granule of G_3 corresponding to the second granule of G_2 of the considered group, starts 1 granules of G_1 after the beginning, and ends 2 granules of G_1 after the ending of the same granule.

C. The Shifting Operation

We define the operation $Shift(G_1, G_2)$ as follows.

Definition 14: Let G_1 be a granularity and m an integer. The granularity G_2 obtained through the algebraic operator $\mathbf{Shift_m}(G_1)$ may be specified by the following PPLTL formula:

$$Shift(G_1, G_2, m) = \mathbf{Gran}(G_1) \wedge \mathbf{Gran}(G_2) \wedge \alpha_m$$

where

$$\alpha_0 = \mathbf{G}((P_{G_1} \leftrightarrow P_{G_2}) \wedge (Q_{G_1} \leftrightarrow Q_{G_2})\wedge (P_{HG_1} \leftrightarrow P_{H_{G_2}}) \wedge (Q_{H_{G_1}} \leftrightarrow Q_{H_{G_2}}))$$

and, for $q > 0$,

$$\alpha_m = \neg(Q_{G_1} \vee P_{G_2})\mathbf{U}(Q_{G_1} \wedge \neg P_{G_2} \wedge \mathbf{X}\alpha_{m-1})$$

□

The first row of the formula requires G_1 and G_2 to be (possibly non internally continuous) granularities. Formula α_m requires that there are no granules of G_2 until the mth granule of G_1; finally, formula α_0 states that, starting from the $(m + 1)$th granule onwards, G_1 and G_2 coincide, including gaps inside granules.

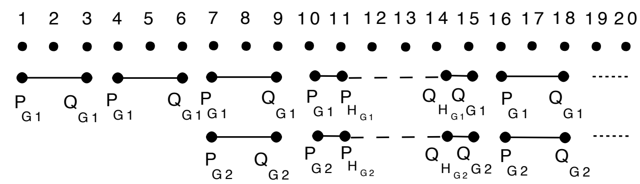

Figure 3. The shifting operation

D. The Combining Operation

We define the operation $Combine(G_1, G_2, G_3)$ as follows.

Definition 15: Let G_1 and G_2 be two granularities. The granularity G_3 obtained through the algebraic operator $\mathbf{Combine}(G_1, G_2)$ may be specified by the following PPLTL formula:

$$\begin{gathered}
Combine(G_1, G_2, G_3) = \\
\mathbf{Gran}(G_1) \wedge \mathbf{Gran}(G_2) \wedge \mathbf{Gran}(G_3) \wedge \\
\mathbf{G}((P_{G_3} \rightarrow \alpha) \wedge (Q_{G_3} \rightarrow \beta))\wedge \\
\mathbf{G}(P_{H_{G_3}} \rightarrow (P_{H_{G_2}} \vee (\neg P_{G_2} \wedge \mathbf{X^{-1}}Q_{G_2})) \wedge \\
Q_{H_{G_3}} \rightarrow (Q_{H_{G_2}} \vee (\neg Q_{G_2} \wedge \mathbf{X}P_{G_2}))).
\end{gathered}$$

where α is

$$(\neg P_{G_1} \wedge \neg Q_{G_1})\mathbf{U}(Q_{G_3} \wedge Q_{G_2}\wedge (Q_{G_1} \vee \mathbf{X}((\neg P_{G_3} \wedge \neg Q_{G_2})\mathbf{U}Q_{G_1}))),$$

and β is

$$(\neg P_{G_1} \wedge \neg Q_{G_1})\mathbf{S}(P_{G_3} \wedge P_{G_2}\wedge (P_{G_1} \vee \mathbf{X^{-1}}((\neg Q_{G_3} \wedge \neg P_{G_2})\mathbf{S}P_{G_1})))$$

□

The first implication ($P_{G_3} \rightarrow \alpha$) states that if a granule of G_1 is included in at least a granule of G_2, then the granule of G_3 that contains it ends with the last granule of G_2 that is included in that granule of G_1.

The second implication ($Q_{G_3} \rightarrow \beta$) states that if a granule of G_1 is included in at least one granule of G_2, then the granule of G_3 containing it begins with the first granule of G_2 that is included in that granule of G_1.

The last two rows of the formula state that a gap in a granule of G_3 may exist only if the same gap exists in the corresponding granule of G_2, or if a gap exists between two granules of G_2 belonging to that granule of G_1. Figure 4 shows an example.

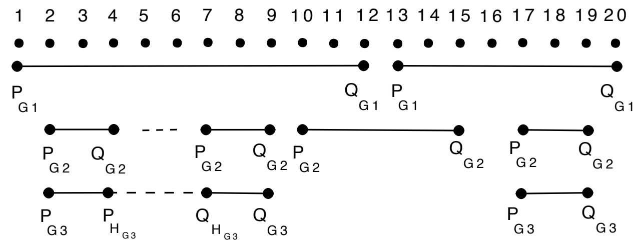

Figure 4. The combining operation

E. The Anchored-grouping Operation

The operation $AnchoredGroup(G_1, G_2, G_3)$ is defined as follows.

Definition 16: Let G_1 and G_2 be two granularities such that $\mathbf{SubGranularity}(G_1,\ G_2)$ holds. The granularity G_3 obtained through the algebraic operator $\mathbf{Anchored\text{-}group}(G_1, G_2)$ may be specified by the following PPLTL formula:

$$\begin{gathered} AnchoredGroup(G_1, G_2, G_3) = \\ \mathbf{Gran}(G_1)\ \wedge \mathbf{Gran}(G_2)\ \wedge \mathbf{Gran}(G_3) \wedge \\ \mathbf{G}(P_{G_1} \to (P_{G_2} \wedge \neg(Q_{G_1} \vee Q_{G_2})\mathbf{U}(Q_{G_1} \wedge Q_{G_2})) \wedge \\ P_{H_{G_1}} \to (P_{H_{G_2}} \wedge \neg(Q_{H_{G_1}} \vee Q_{H_{G_2}})\mathbf{U}(Q_{H_{G_1}} \wedge Q_{H_{G_2}}))) \wedge \\ \mathbf{G}(P_{H_{G_3}} \to (P_{H_{G_1}} \vee (\neg P_{G_1} \wedge \mathbf{X^{-1}} Q_{G_1})) \wedge \\ Q_{H_{G_3}} \to (Q_{H_{G_1}} \vee (\neg Q_{G_1} \wedge \mathbf{X} P_{G_1}))) \wedge \\ \mathbf{G}(P_{G_3} \to (P_{G_2} \wedge \mathbf{X}((\neg P_{G_2} \wedge \neg Q_{G_3})\mathbf{U}(Q_{G_3} \wedge \mathbf{X} P_{G_2}))). \end{gathered}$$

□

The second and the third row of the formula state that each granule of G_2 has to be equal to a granule of G_1, and possible gaps inside a granule of G_2 has to be equal to the gaps inside the corresponding granule of G_1 (i.e., that the relationship $\mathbf{SubGranularity}(G_1, G_2)$ holds). The fourth and the fifth rows state that a granule of G_3 may have gaps only if G_1 has the same gaps inside its granules, or corresponding to a gap between two non consecutive granuels of G_1. Finally, the last row states that a ganule of G_3 begins at the same time instant of a granule of G_2, and ends at the time instant that precedes the beginning of the next granule of G_2. Figure 5 shows an example.

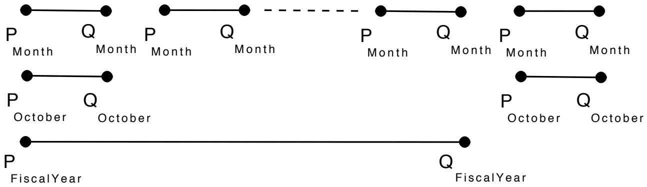

Figure 5. The anchored grouping operation

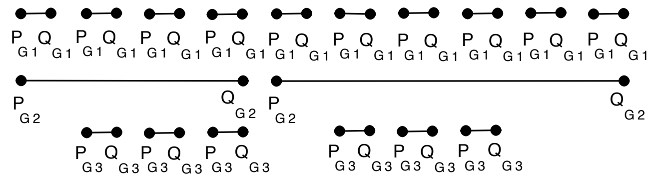

Figure 6. The Select-down operation

F. The Select-down Operation

The operation $SelDown^l_k(G_1, G_2, G_3)$ is defined as follows.

Definition 17: Let G_1 and G_2 be two granularities, and k, l two integers. The granularity G_3 obtained through the algebraic operator $\mathbf{Select\text{-}down}^l_k(G_1, G_2)$ may be specified by the following PPLTL formula:

$$\begin{gathered} SelDown^l_k(G_1, G_2, G_3) = \mathbf{Partition}(G_1, G_2) \wedge \\ \mathbf{Subgranularity}(G_1, G_3)\ \wedge P_{G_2} \to \alpha_k \end{gathered}$$

where

$$\begin{aligned} \alpha_0 &= P_{G_1} \wedge P_{G_3} \wedge \beta_l \\ \alpha_k &= (\neg P_{G_1} \wedge \neg P_{G_3})\mathbf{U}(P_{G_1} \wedge \mathbf{X}\alpha_{n-1}) \\ \beta_0 &= \neg P_{G_3}\mathbf{U} Q_{G_2} \\ \beta_n &= (\neg P_{G_1} \wedge \neg P_{G_3})\mathbf{U}(P_{G_1} \wedge P_{G_3} \wedge \mathbf{X}\beta_{n-1}) \end{aligned}$$

□

The third conjunct of the formula ($P_{G_2} \to \alpha_k$) considers each granule of G_2 separately; the formula α_k identifies the k^{th} granule of G_1 in the group of granules of G_1 that are included in the considered granule of G_2; the formula β_l specifies that, for the subsequent l granules of G_1, a corresponding granule of G_3 exists. Figure 6 shows an example where $k = 2$ and $l = 3$.

G. A Clinical Example

In order to show the benefits of the proposed logical specifications of the algebraic operations, we consider a real-world example taken from the clinical scenario. The example deals with one of the several chemotherapy recommendations that can be represented through a set of time granularities, and models these time granularities in a labelled linear time structure, identified by a suitable PPLTL formula. The chemotherapy recommendation is defined as follows: "*The ChlVPP regimen consists of chlorambucil (6 mg/m^2/day) on days 1 through 14, vinblastine (6 mg/m^2) on days 1 and 8, procarbazine (100 mg/m^2/day) on days 1 through 14, and prednisone (40 mg/day) on days 1 through 14. Patients treated with this regimen receive 6 cycles repeated every 28 days.*" Figure 7 shows a graphical representation of a single cycle of the regimen.

The labelled linear time structure corresponding to the granularities representing the chemotherapy-related granu-

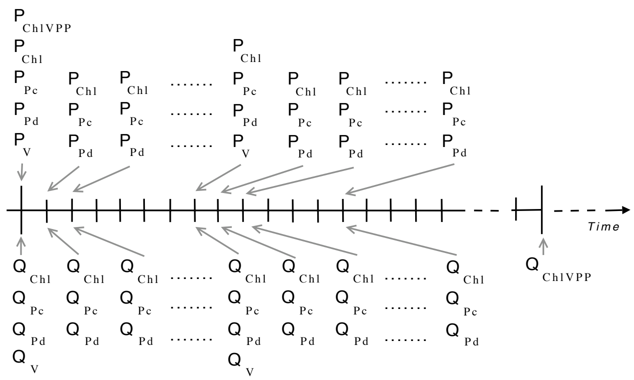

Figure 7. ChlVPP chemotherapy regimen

larities may be expressed as:

$Count(P_{\mathsf{ChDay}}, 168) \wedge Subgranularity(\mathsf{Chday}, \mathsf{day}) \wedge$
$Group(\mathsf{ChDay}, \mathsf{ChlVPP}, 28) \wedge SelDown_1^{14}(\mathsf{day}, \mathsf{ChlVPP}, \mathsf{Chl}) \wedge$
$SelDown_1^1(\mathsf{day}, \mathsf{ChlVPP}, \mathsf{V}) \wedge SelDown_8^1(\mathsf{day}, \mathsf{ChlVPP}, \mathsf{V}) \wedge$
$SelDown_1^{14}(\mathsf{day}, \mathsf{ChlVPP}, \mathsf{Pc}) \wedge SelDown_1^{14}(\mathsf{day}, \mathsf{ChlVPP}, \mathsf{Pd})$

The above formula defines granularities Chl (chlorambucil), V (vinblastine), Pc (procarbazine), Pd (prednisone) and ChlVPP according to the recommendation; the formula is defined on the time domain $\mathbb{N}$ of days (we assume that the granularity day is composed of a granule for each point in the considered time domain). The first two conjuncts state that the granularity ChDay consists of 128 days, which is the overall duration of 6 cycles: $Count(\alpha, n)$ stands for "α holds exactly n times in the future" and is defined as follows [3]: $Count(\alpha, 0) = \mathbf{G}\neg\alpha$ and $Count(\alpha, n) = \neg\alpha\mathbf{U}(\alpha \;\&\; \mathbf{X}Count(\alpha, k-1))$. The next conjunct states that each granule of ChlVPP is of 28 elements (days of chemotherapy). The next conjunct states that the granularity Chl covers the first 14 days of each cycle; the next two conjuncts state that the granularity V covers the first and the eighth days of each cycle; finally, the last two conjuncts state that the granularities Pc and Pd cover the first 14 days of each cycle. The adoption of the logical specification of the grouping and select-down operations makes the overall formula more natural and intuitive than the corresponding one specified only through the PPLTL syntax.

IV. Discussion and Conclusions

In this paper we considered the logical framework of Combi *et al.* [3] for representing and reasoning about temporal granularities, and the algebraic framework of Ning *et al.* [7]. Among the approaches for symbolic specification of granularities, such as those proposed by Foster, Leban, and McDonald [5] and by Niezette and Stevenne [6], we focused on the calendar algebra, which is, differently from the other proposals, based on the same general framework for time granularities [1] adopted by the considered logical approach. Moreover, calendar algebra provides users with a large set of operators simplifying the definition of even complex granularities. We proposed logical formulae encoding grouping-oriented operators of the calendar algebra [7] such as *grouping*, *altering-tick*, *shifting*, *combining* and *anchored-grouping*, and the granule-oriented operation *select-down*. The logical formula encoding the *altering-tick* operator performs the operation only on the first p groups of m-granules of the first granularity, and deals with internally continuous granularities. The remaining logical formulae deal even with non internally continuous granularities. Founding the operators of the calendar algebra on a logical framework allow us to use all the machinery studied and developed for temporal logics; as an example, in the considered medical domain it could be of interest to instantiate some chemotherapy cycles. It corresponds to the generation of granularities, that can be reduced to the problem of generating models for the formula corresponding to chemotherapy-related granularities [3]. With an approach similar to that we proposed in this paper, in [2] efficient conversion methods are defined from an algebraic granularity representation to the equivalent low-level representation based on periodical sets, where many reasoning tools are available. Even though the final goal of this proposal is similar to that of our proposal, it is worth noting that we do not provide conversion methods to a lower level representation: indeed, we provide within the logical framework based on PPLTL some "formula constructors" which allow one to specify in an algebraic-like concise fashion possibly long and not easily readable logical formulae defining temporal granularities.

References

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

[2] C. Bettini, S. Mascetti, and X. S. Wang. Supporting temporal reasoning by mapping calendar expressions to minimal periodic sets. *Journal of Artificial Intellicence Research*, 28:299–348, 2007.

[3] C. Combi, M. Franceschet, and A. Peron. Representing and reasoning about temporal granularities. *Journal of Logic and Computation*, 14(1):51–77, 2004.

[4] S. Demri. LTL over integer periodicity constraints. *Theoretical Compututer Science*, 360(1-3):96–123, 2006.

[5] D. Foster, B. Leban, and D. McDonald. A representation for collections of temporal intervals. In *Proceedings of the American National Conference on Artificial Intelligence*, pages 367–371, 1986.

[6] M. Niezette and J. Stevenne. An efficient symbolic representation of periodic time. In *Implementing Temporal Reasoning: workshop notes - AAAI'92*, pages 130–140, 1992.

[7] P. Ning, S. Jajodia, and X. S. Wang. An algebraic representation of calendars. *Annals of Mathematics and Artificial Intelligence*, 36:5–38, 2002.