A logical approach to represent and reason about calendars

Carlo Combi
Department of Computer Science, University of Verona, Italy
E-mail: combi@sci.univr.it

Massimo Franceschet
Department of Sciences, University 'G. D'Annunzio' of Pescara, Italy
E-mail: francesc@sci.unich.it

Adriano Peron
Department of Physical Sciences, University 'Federico II' of Napoli, Italy
E-mail: peron@na.infn.it

Abstract

In this paper, we propose a logical approach to represent and reason about different time granularities. We identify a time granularity as a discrete infinite sequence of time points properly labelled with proposition symbols marking the starting and ending points of the corresponding granules, and we intensively model sets of granularities with linear time logic formulas. Some real-world granularities are provided, to motivate and exemplify our approach. The proposed framework permits to algorithmically solve the consistency, the equivalence, and the classification problems in a uniform way, by reducing them to the validity problem for the considered linear time logic.

1. Introduction

The ability of providing and relating temporal representations at different 'grain levels' of the same reality is an important research theme in computer science and a major requirement for formal specifications, temporal databases, data mining, problem solving, and natural language understanding [1].

Any time granularity can be viewed as the partitioning of a temporal domain in groups of elements, where each group is perceived as an indivisible unit (a granule). The description of a fact can use these granules to provide it with a temporal qualification, at the appropriate abstraction level. In order to represent and reason about time granularity, any formalism should satisfy the following requirements:

- *Expressiveness*. The class of granularities represented in the formalism should be large enough to be of practical use.
- *Effectiveness*. The framework should provide algorithms to reason about different time granularities. In particular, it should provide an effective solution to the well-known problems of *consistency*, *equivalence* and *classification*. The *consistency problem* is the problem of deciding whether a granularity representation is well-defined. The algorithmic solution of the consistency problem is important to avoid the definition of inconsistent granularities that may produce unexpected failures in the system. The *equivalence problem* is the problem of deciding whether two different representations define the same granularity. The decidability of the equivalence problem implies the possibility of effectively testing the semantic equivalence of two different time granularity representations, making it possible to use the smaller and more tractable one. The *classification problem* solves the problem of deciding whether a natural number n belongs to a granule of a given granularity. The classification problem is strictly related to the granule conversion problem which allows one to relate granules of a given granularity to granules of another one.
- *Compactness*. The formalism should exploit regularities exhibited by the considered granularities to make their representations as compact as possible.

The frameworks to represent and reason about time granularity present in the literature can be classified into algebraic frameworks and logical frameworks. Algebraic frameworks for time granularities have been proposed by Ning, Jajodia and Wang [13], Foster, Leban and McDonald [5], and Niezette and Stevenne [12]. In an *algebraic* (or *operational*) framework, a *bottom granularity* is assumed, and a finite set of *calendar operators* are exploited to create new granularities by suitably manipulating other granularities. A granularity is hence identified by an algebraic expression. Logical approaches to represent and reason about time granularity, based on a many-level view of temporal structures, have been proposed by Montanari in [8], and further investigated by Franceschet, Montanari, Peron and Policriti in [6, 9, 10, 11]. In a *logical* (or *descriptive*) framework for time granularity, the different granularities and their interconnections are represented by means of mathematical structures called layered structures, consisting of a possibly infinite set of related differently-grained temporal domains. Suitable operators make it possible to move horizontally *within* a given temporal domain (displacement operators), and to move vertically *across* temporal domains (projection operators). Logical formulas allow one to specify properties involving different time granularities in a single formula by mixing displacement and projection operators.

A comparison of the algebraic and the logical frameworks is not immediate. The main reason is that these frameworks have been applied to different application fields calling for different requirements. For instance, in the database context, granule conversion plays a major role because it allows the user to view the temporal information contained in the database in terms of different granularities, while in the context of verification, decision procedures for consistency and model checking are unavoidable to validate the system. However, abstracting away from the application fields of the two frameworks, a comparison is possible. The main advantage of the algebraic framework is its naturalness: by applying user-friendly operations to existing standard granularities like ‘days’, ‘weeks’, and ‘months’, a quite large class of new granularities, like ‘business weeks’, ‘business months’, and ‘years since 2000’, can be easily generated. The major weakness of the algebraic approach is that reasoning methods basically reduce to granule conversions and semantic translations of statements. Little attention has received the investigation on algorithms to check whether some meaningful relation holds between granularities (e.g., to verify whether the granularity G_1 is finer than granularity G_2 or G_1 is equivalent to G_2). Moreover, only a finite number of time granularities can be represented. On the contrary, reasoning methods have been extensively investigated in the logical framework, where both a finite and an infinite number of time granularities can be dealt with. Theorem provers make it possible to verify whether a granular requirement is consistent, while model checkers allow one to check whether a granular property is satisfied in a particular structure. To allow such computational properties, however, some assumptions have to be made about the involved granularities, as, for example, some form of regularity of the sizes of the granules.

In this paper, we propose a logical approach to represent and reason about different time granularities. We identify a time granularity with a discrete linear time structure properly labelled with proposition symbols marking the starting and ending points of the corresponding granules. We make use of a linear time logic, interpreted over labelled linear time structures, to model possibly infinite sets of time granularities. Any linear time formula is associated with a set of labelled linear time structures satisfying the formula (the set of models of the formula). Since any properly labelled linear time structure identifies a time granularity, we may model possibly infinite sets of time granularities by means of well-defined linear time formulas. Moreover, a single sequence may identify a finite number of different granularities (a calendar) by using a different couple of marking proposition symbols for any granularity. Hence, well-defined linear time formulas may model possibly infinite sets of calendars as well. The proposed framework permits to algorithmically solve the consistency, the equivalence, and the classification problems in a uniform framework by reducing them to the validity problem for the considered linear time logic, which is known to be decidable in polynomial space. Our approach is logical, and hence it intrinsically differs from the algebraic one. However, the starting point of our approach and of the algebraic one is the same: the classical definition of time granularity given in [1]. Moreover, our logical approach differs from the above described logical one of Montanari et al for the following reason. While Montanari et al model different time granularities by using multi-layered mathematical structures and use temporal logic formulas to capture *properties* of time granularities, we model both time granularities and their properties by using temporal logic formulas. Our solution enhances the flexibility of the task of granularity specification: the time granularity structure may be changed by simply modifying the logical formula that defines it, and the properties of the time granularity structure may be defined in the same logical language.

The rest of the paper is as follows. In Section 2 we present some motivating examples. In Section 3 we propose our logical approach to represent and reason about time granularity, while in Section 4 we compare our work with related ones and we outline future work.

2. Motivating examples

In introducing the needs of managing different granularities, we will focus on an example coming from clinical medicine. In particular, we focus on the definition of specific granularities related to therapy plans. Intuitively, therapy plans can be considered as the calendar according to which it is possible to properly observe the evolution of the patient's state.

We consider here chemotherapies for oncological patients, a topic which has been extensively considered by the clinical research and that is precisely described and recommended in several clinical practice guidelines. In general, oncology patients undergo several chemotherapy cycles: each cycle can include the administration of several drugs, which the patient has to assume according to a specific temporal pattern.

As an example, consider the following chemotherapy recommendation [7]:

> "The recommended CMF[1] regimen consists of 14 days of oral cyclophosphamide with intravenous methotrexate, and 5-fluorouracil on days 1 and 8. This is repeated every 28 days for 6 cycles."

Moreover, it may happen that the beginning of a cycle is delayed a few days, due to patient's blood analysis results. Figure 1 provides a graphical representation of the recommended CMF regimen. According to this scenario, we can easily identify some requirements related to the definition of useful granularity systems:

1. *Definition of therapy-related granularities*. These granularities should be suitably specified for different patients, according to the moment at which they start a given chemotherapy.

2. *Definition of granularities having some degree of uncertainty*. There is, indeed, the need of representing the fact that two consecutive cycles may be separated by some days, due to the patient's conditions.

3. *Verification of consistency between an assigned therapy and the recommended regimen*. Given a therapy assigned to a patient with the specification of days and corresponding drug assumptions, it is important to be able to determine whether the therapy is consistent with the recommended regimen.

4. *Assignment of a therapy according to the recommended regimen and to other granularity-related constraints*. It could be necessary, for organizational reasons, to avoid that some specific drug administrations happen during the weekend: for example, in specifying a CMF therapy, we could avoid that the administration of 5-fluorouracil is on Sundays.

3. A logical approach to represent and reason about calendars

In this section, we propose our logical approach to represent and reason about different time granularities.

3.1. Representing time granularity

We model time granularity according to the following definition.

Definition 3.1 *A* granularity *is a mapping $G : \mathbb{N} \rightarrow 2^{\mathbb{N}}$ such that:*

1. *for all $i < j$, for any $n \in G(i)$ and $m \in G(j)$, $n < m$;*

2. *for all $i < j$, if $G(j) \neq \emptyset$, then $G(i) \neq \emptyset$;*

3. *for any $i \in \mathbb{N}$, $G(i)$ is a convex interval.*

Following the classical definition given in [1], the domain of a granularity G is called *index set* and an element of the codomain of G is called a *granule*. The definition of granularity above specializes the definition given in [1], assuming that both the index set and the domain of granules are the linear discrete domain $(\mathbb{N}, <)$. The first condition states that granules in a granularity do not overlap and that their index order is the same as their time domain order. The second condition states that the subset of the index set that maps to nonempty granules forms an initial segment. The third condition avoids granularities with gaps inside the granules (this assumption will be relaxed in the following).

Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ be a *finite* set of granularities (we will refer to $\mathcal{G}$ as a *calendar*), and let $\mathcal{P}_{\mathcal{G}} = \{P_{G_i}, Q_{G_i} \mid 1 \leq i \leq n\}$ be a set of proposition symbols associated with the calendar $\mathcal{G}$. Given an alphabet of proposition symbols $\mathcal{P} \supseteq \mathcal{P}_{\mathcal{G}}$, we shall consider in the following $\mathcal{P}$-labelled (discrete) linear time structures having the form $(\mathbb{N}, <, V)$, where $(\mathbb{N}, <)$ is the set of natural numbers with the usual ordering, and $V : \mathbb{N} \rightarrow 2^{\mathcal{P}}$ is a labelling function mapping natural numbers to sets of proposition symbols. The idea is to identify a time granularity G, according to Definition 3.1, with a linear time structure, properly labelled with proposition symbols taken from $\{P_G, Q_G\}$: the starting (resp. ending) point of an arbitrary granule of G in the structure is labelled by P_G (resp. Q_G).

Definition 3.2 *A labelled linear time structure $(\mathbb{N}, <, V)$ is G-consistent whenever:*

- *if $P_G \in V(i)$ for some $i \in \mathbb{N}$, then either $Q_G \in V(i)$ or $Q_G \in V(j)$ for some $j > i$ such that $P_G \notin V(k)$ for each $i < k \leq j$ and $Q_G \notin V(k)$ for each $i \leq k < j$;*

[1] CMF stands for the chemotherapy based on the drugs Cyclophosphamide, Methotrexate, and 5-Fluorouracil.

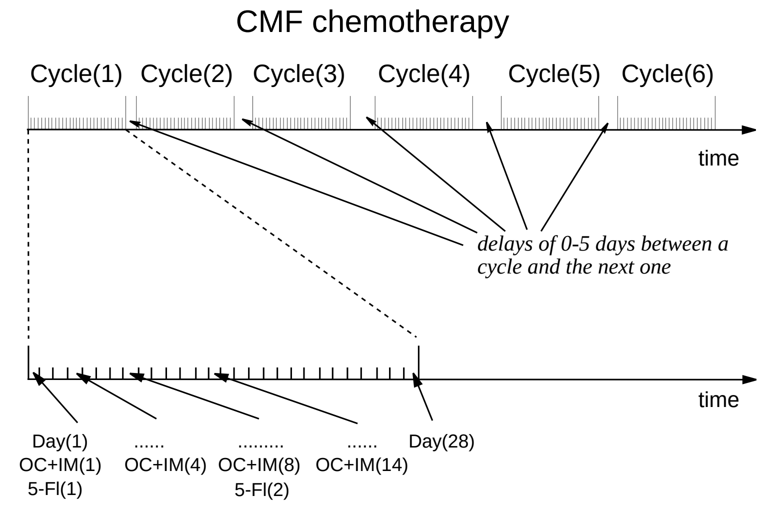

Figure 1. Granularities involved in a chemotherapy treatment.

- *if* $Q_G \in V(i)$ *for some* $i \in \mathbb{N}$, *then either* $P_G \in V(i)$ *or* $P_G \in V(j)$ *for some* $j < i$ *such that* $Q_G \notin V(k)$ *for each* $j \leq k < i$ *and* $P_G \notin V(k)$ *for each* $j < k \leq i$.

The above conditions say that every point labelled with P_G has to match with a unique point labelled with Q_G, and vice versa. It is easy to see that every G-consistent labelled linear time structure induces a granularity G: given a G-consistent labelled linear time structure $\mathcal{M} = (\mathbb{N}, <, V)$, a granule of $\mathcal{M}$ with respect to G is a set $\{n, n+1, \ldots, n+k\}$, for some $n, k \geq 0$, such that $P_G \in V(n)$, $Q_G \in V(n+k)$ and $Q_G \notin V(n+j)$ for all $0 \leq j < k$. The granularity G induced by $\mathcal{M}$ is such that $G(i)$ is the i-th granule of $\mathcal{M}$ with respect to G, if any, and $G(i) = \emptyset$ otherwise. Similarly, a granularity G induces a G-consistent labelled linear time structure.

Example 3.3 We give two examples of labelled linear time structures that induce well-defined granularities and one example of a labelled linear time structure that does not correspond to a granularity.

- The structure $(\mathbb{N}, <, V)$ such that $V(i) = \{P_G\}$ iff i is even, and $V(i) = \{Q_G\}$ iff i is odd, induces the uniform, continuous and total granularity G such that $G(i) = \{2 \cdot i, 2 \cdot i + 1\}$ for any $i \in \mathbb{N}$.
- The structure $(\mathbb{N}, <, V)$ such that $V(0) = \{P_G\}$, $V(1) = \{Q_G\}$, $V(3) = \{P_G\}$, $V(5) = \{Q_G\}$ induces the non-uniform, non-continuous, non-total granularity G such that $G(0) = \{0, 1\}$, $G(1) = \{3, 4, 5\}$, and $G(i) = \emptyset$ for each $i \geq 2$.
- The structure $(\mathbb{N}, <, V)$ such that $V(0) = \{P_G\}$, $V(1) = \{Q_G, P_G\}$, $V(2) = \{Q_G\}$ does not induce any granularity, since it is not G-consistent (indeed, the granules $G(0) = \{0, 1\}$ and $G(1) = \{1, 2\}$ intersect).

In the following we show how a set of granularities can be defined in an intensional declarative manner by means of a formula of a propositional linear time logic (instead of defining it extensively as done in Example 3.3). We will use Past Propositional Linear Time Logic (PPLTL for short) [4], interpreted over labelled linear time structures. A PPLTL-formula intensionally defines a possibly infinite set of labelled linear time structures, which correspond to the linear time structures satisfying the formula. Since, as shown above, consistently labelled linear time structures correspond to granularities, we can use suitable linear time formulas to define sets of granularities. We proceed by introducing the syntax and the semantics of PPLTL.

Definition 3.4 (*Syntax of* PPLTL)

Formulas of PPLTL *are inductively defined as follows:*

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

Formulas $\phi \vee \psi$, $\phi \rightarrow \psi$, and $\phi \leftrightarrow \psi$ are defined as $\neg(\neg\phi \wedge \neg\psi)$, $\neg\phi \vee \psi$, and $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, respectively. Moreover, $\mathbf{F}p$ (p will hold in the future), $\mathbf{G}p$ (p will always hold in the future), $\mathbf{P}p$ (p held in the past) and $\mathbf{H}p$ (p always held in the past) are shorthands for, respectively, $\mathtt{true}\mathbf{U}p$, $\neg\mathbf{F}\neg p$, $\mathtt{true}\mathbf{S}p$ and $\neg\mathbf{P}\neg p$, where $\mathtt{true} = P \vee \neg P$, for some $P \in \mathcal{P}$.

We interpret PPLTL over $\mathcal{P}$-labelled linear time structures. The semantics of PPLTL is as follows.

Definition 3.5 (*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 ψ in $\mathcal{M}$ with respect to the point i, denoted $\mathcal{M}, i \models \psi$, is defined as follows:*

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

We say that $\mathcal{M}$ is a model of ψ if $\mathcal{M}, 0 \models \psi$.

Example 3.6 We give some examples of how PPLTL-formulas can encode sets of granularities.

- The set of all G-consistent granularities (according to Definition 3.2) is captured by the following PPLTL-formula

  $$\eta(G) = \mathbf{G}((P_G \to \alpha) \wedge (Q_G \to \beta)),$$

  where

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

- The (singleton containing the) granularity G such that $G(i) = \{2 \cdot i, 2 \cdot i + 1\}$, for each $i \in \mathbb{N}$, is defined by the following formula:

  $$\eta(G) \wedge P_G \wedge \mathbf{G}(P_G \to (\neg Q_G \wedge \neg\mathbf{X}P_G \wedge \mathbf{XX}P_G)).$$

- The infinite set of granularities obtained by an arbitrarily right-shifting G (i.e., the non-anchored version of G), where $G(i) = \{2 \cdot i, 2 \cdot i + 1\}$, for each $i \in \mathbb{N}$, is encoded by the following formula:

  $$\begin{array}{l}
  \eta(G) \wedge \mathbf{F}(P_G \wedge \mathbf{G}(P_G \to \\
  (\neg Q_G \wedge \neg\mathbf{X}P_G \wedge \mathbf{XX}P_G))).
  \end{array}$$

A finite number of different granularities may be addressed in the same formula by using different pairs of marking proposition symbols. For instance, given a calendar $\mathcal{G} = \{G_1, \ldots, G_n\}$, the formula $\bigwedge_{i=1}^{n} \eta(G_i)$ defines the set of all calendars with n granularities $G_1, \ldots, G_n$. Meaningful relations between granularities belonging to the calendar $\mathcal{G}$ may be captured in our framework. For instance, G_1 `GroupInto` G_2 (each granule of G_2 is obtained by grouping granules of G_1), G_1 `FinerThan` G_2 (each granule of G_1 is contained in some granule of G_2), G_1 `SubGranularityOf` G_2 (each granule of G_1 is equal to a granule of G_2) [13]. As an example, the relation `FinerThan` can be captured by the following formula:

$$\eta(G_1) \wedge \eta(G_2) \wedge \mathbf{G}((P_{G_1} \to \alpha) \wedge (Q_{G_1} \to \beta)),$$

where

$$\begin{array}{lll}
\alpha & = & \neg(P_{G_2} \vee Q_{G_2})\mathbf{U}(Q_{G_1} \wedge \\
 & & (Q_{G_2} \vee \mathbf{X}(\neg P_{G_2}\mathbf{U}Q_{G_2}))) \\
\beta & = & \neg(P_{G_2} \vee Q_{G_2})\mathbf{S}(P_{G_1} \wedge \\
 & & (P_{G_2} \vee \mathbf{X^{-1}}(\neg Q_{G_2}\mathbf{S}P_{G_2})))
\end{array}$$

Hence, the calendars with n granularities that are totally ordered with respect to the `FinerThan` relation are defined by the following formula:

$$\bigwedge_{i=1}^{n} \eta(G_i) \wedge \bigwedge_{i=1}^{n-1} \mathtt{FinerThan}(G_i, G_{i+1}).$$

The above framework does not consider granularities with gaps inside the granules (only convex granules are treated). However, it can be easily extended to cope with non-convex granules. The idea is to use symbols P_G and Q_G to delimit the granules of a granularity G as done before, and symbols P_{H_G} and Q_{H_G} to bound the gaps inside the granules of G. In this way we have that the description of the gaps of G is itself a granularity H_G. Note that H_G is finer than G. Indeed, every internal gap of G (a granule of H_G) is a subset of some granule of G. Moreover, there are no granules of G that are entirely covered by gaps (granules of H_G): H_G `GroupInto` G does not hold.

We conclude this section by reconsidering the chemotherapy treatment described in Section 2.

Example 3.7 Let us assume that OC (cyclophosphamide), IM (intravenous methotrexate) and FI (5-fluorouracil) are proposition symbols corresponding to the drugs of the CMF regimen. We preliminary introduce some useful shorthands. For a formula p, $\mathbf{X}^n(p)$ stands for "p holds in n time instants", and is defined as follows: $\mathbf{X}^0(p) = p$, and $\mathbf{X}^n(p) = \mathbf{X}\mathbf{X}^{n-1}(p)$. For $0 \leq n \leq m$, $\forall[n,m](p)$ (resp. $\exists[n,m](p)$) stands for "p holds everywhere (resp. somewhere) in the time interval $[n,m]$", and is defined as $\bigwedge_{i=n}^{m} \mathbf{X^i}(p)$ (resp. $\bigvee_{i=n}^{m} \mathbf{X^i}(p)$). Finally, $Count(p,n)$ stands for "p holds exactly n times in the future" and is defined as follows: $Count(p,0) = \mathbf{G}(\neg p)$ and $Count(p,n) = \neg p\mathbf{U}(p \wedge \mathbf{X}(Count(p,k-1)))$.

The formula Ω_{CMF} below defines, on the time domain $\mathbb{N}$ of days, a granularity CMF according to the recommendation given in Section 2:

$$\begin{aligned}&\eta(CMF) \wedge \\ &Count(P_{CMF}, 6) \wedge \\ &\mathbf{G}(P_{CMF} \rightarrow (\forall[0,26](\neg Q_{CMF}) \wedge \mathbf{X}^{27} Q_{CMF})) \wedge \\ &\mathbf{G}((Q_{CMF} \wedge \mathbf{F} P_{CMF}) \rightarrow \exists[1,5] P_{CMF}) \wedge \\ &\mathbf{G}(P_{CMF} \rightarrow (\forall[0,13](OC \wedge IM) \wedge FI \wedge \mathbf{X}^{7}(FI) \wedge \\ &\forall[14,27](\neg OC \wedge \neg IM \wedge \neg FI)))\end{aligned}$$

The first conjunct says that CMF is a granularity. The second and the third conjuncts say, respectively, that the granularity CMF consists of 6 granules (cycles) each of 28 elements (days). The fourth conjunct states that each cycle is separated by time intervals not exceeding 5 units. Finally, the fifth conjunct associates the drugs to each day in the cycle, according to the recommendation (the first 14 days cyclophosphamide and intravenous methotrexate, with 5-fluorouracil on days 1 and 8, and no drugs during the second 14 days).

It is worth noting that in the above example only a bounded form of uncertainty is involved. Indeed, two successive cycles may be separated by no more than 5 time units (in the chosen granularity). However, there exist applications calling for *unbounded uncertainty*. For instance, two therapy cycles that are arbitrarily distant. Our framework can cope with unbounded uncertainty as well.

3.2. Reasoning about time granularity

Besides representing sets of granularities and relations among them, our framework permits to automatically solve the following problems.

Automatic verification of granularity properties. For verifying properties against models, we can exploit in a natural way the mature technology of *model checking* [2]. A labelled structure representing a concrete set of granularities can be suitably encoded in the format required by the chosen model checker and it can be checked against a formula of a propositional linear logic which describes the required property. For instance, with reference to our clinical example, a concrete chemotherapy plan can be checked for consistency against the formula Ω_{CMF} describing the chemotherapy regimen (as described in the requirement 3 of Section 2).

Automatic generation of granularities. Given a formula φ defining a set of granularities, it it possible to automatically generate the labelled linear time structures satisfying φ. More technically, it is possible to construct a finite state automaton which accepts the set of structures satisfying the formula φ. Any concrete labelled linear time structure satisfying the formula φ can be obtained by suitably unravelling the automaton. For instance, we can obtain all of the possible schedules for a chemotherapy according to the regimen encoded by the formula Ω_{CMF}.

Consistency, equivalence and classification. As far as consistency is concerned, given a PPLTL-formula φ, one can verify whether it encodes a set of well-defined granularities (according to Definition 3.1) by checking the validity of the formula $\varphi \rightarrow \eta(G)$.

The equivalence problem for two sets of granularities defined by φ_1 and φ_2, respectively, is reduced to checking the validity of the formula $\varphi_1 \leftrightarrow \varphi_2$.

Finally, the classification problem, that is, the problem of checking whether a natural number n belongs to a granule of a granularity G defined by a PPLTL-formula φ can be solved as follows. We have that $n \geq 0$ is contained in some granule of any granularity defined by φ if the formula $\varphi \rightarrow \alpha_n(G)$ is valid, where $\alpha_n(G)$ is the formula:

$$\begin{aligned}&\mathbf{X^n}(P_G \vee Q_G) \vee \\ &\mathbf{X^n}(\neg(P_G \vee Q_G)\mathbf{S}P_G \wedge \neg(P_G \vee Q_G)\mathbf{U}Q_G)\end{aligned}$$

It is worth pointing out that the model checking and the validity problems for linear time logics have been extensively studied. Both the problems belong to the complexity class PSPACE (polynomial space) [4], and efficient procedures for solving them are at hand [2].

4. Related and future work

A related recent approach to represent and reason about time granularity has been proposed by Wijsen [14] and refined by Dal Lago and Montanari [3]. Wijsen modelled infinite granularities as infinite strings over a suitable finite alphabet. The resulting string-based model is then used to formally state and solve problems of granularity equivalence and minimality. Dal Lago and Montanari gave an automata-theoretic counterpart of the string-based model. They used single string automata, that is, finite-state automata accepting a single infinite string, to represent in a compact way time granularities and to give an algorithmic solution to the problems of equivalence and classification of time granularities. The resulting formalism satisfies the requirements of effectiveness and compactness. As for expressiveness, it is not able to represent *dynamic granularities*, that is, granularities that are not anchored to the underlying time domain. A typical example of dynamic granularity is a repeating pattern that can start at an arbitrary time point. Our formalism has the expressive power to represent both static and dynamic granularities. Dynamic ones are encoded by temporal formulas representing a possibly infinite set of granularities. Moreover, our formalism is effective: consistency, equivalence, and classification problems can be algorithmically solved in a uniform and elegant way. However, our formalism lacks compactness: the representation formula can be very long whenever a periodic granularity has a long prefix or period (for instance, in the case of the Gregorian calendar). We aim at finding a possible way out to this problem within our logical framework. A solution may reside in the use of a *metric* linear time logic, which extends linear time logic with methods to express *quantitative* temporal requirements.

References

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

[2] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999.

[3] U. Dal Lago and A. Montanari. Calendars, time granularities, and automata. In *Proceedings of the International Symposium on Spatial and Temporal Databases*, volume 2121 of *Lectures Notes on Computer Science*, pages 279–298, Los Angeles, CA, USA, 2001.

[4] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. B*, pages 995–1072. Elsevier Science Publishers B.V., 1990.

[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. Franceschet and A. Montanari. Branching within time: an expressively complete and elementarily decidable temporal logic for time granularity. *Journal of Language and Computation*. To appear.

[7] M. Levine, C. Sawka, and D.M. Bowman. Clinical practice guidelines for the care and treatment of breast cancer: 8. adjuvant systemic therapy for women with node-positive breast cancer (2001 update). *Canadian Medical Association Journal*, 164, 2001.

[8] A. Montanari. *Metric and Layered Temporal Logic for Time Granularity*. ILLC Dissertation Series 1996-02, Institute for Logic, Language and Computation, University of Amsterdam, 1996.

[9] A. Montanari, A. Peron, and A. Policriti. Extending Kamp's theorem to model time granularity. *Journal of Logic and Computation*. To appear.

[10] A. Montanari, A. Peron, and A. Policriti. Decidable theories of ω-layered metric temporal structures. *Logic Journal of the IGPL*, 7(1):79–102, 1999.

[11] A. Montanari and A. Policriti. Decidability results for metric and layered temporal logics. *Notre Dame Journal of Formal Logic*, 37:260–282, 1996.

[12] M. Niezette and J. Stevenne. An efficient symbolic representation of periodic time. In *Proceeding of the International Conference on Information and Knowledge Management*, volume 752 of *Lecture Notes in Computer Science*, pages 161–168, Baltimore, Maryland, November 1993.

[13] P. Ning, S. Jajodia, and X. S. Wang. An algebraic representation of calendars. *Annals of Mathematics and Artificial Intelligence*. To appear.

[14] J. Wijsen. A string based-model for infinite granularities. In *Proceedings of the AAAI Workshop on Spatial and Temporal Granularity*, pages 9–16. AAAI Press, 2000.