Periodicity-based temporal constraints

Paolo Terenziani*, Luca Anselma^o, Stefania Montani*

*DI, Univ. Piemonte Orientale “A. Avogadro”, Spalto Marengo 33, Alessandria, Italy
{terenz, stefania}@mf.unipmn.it

^oDI, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy
anselma@di.unito.it

Keywords: User-defined periodicities. Periodicity-based temporal constraints. Temporal constraint propagation. Intensional and extensional approach.

Abstract

In this paper, we propose a framework for representing and reasoning about qualitative and quantitative temporal constraints on periodic events. We describe a formalism to deal with both qualitative and quantitative “periodicity-dependent” constraints between repeated events, considering also user-defined periodicities. We first propose a correct but not complete intensional approach to temporal reasoning based on the operations of intersection and composition, and then we complement it with a correct and complete extensional approach.

1 Introduction

Temporal representation and temporal reasoning are fundamental issues within the AI community, since they play a central role in many “intelligent” activities. Starting from the early 80’s, many AI researchers took interest into the development of specialised approaches to deal with different kinds of temporal constraints [Vila, 94]. In particular, a relevant amount of work has been devoted to the treatment of (constraints between) periodic events (consider, e.g., [Ladkin, 86; Ligozat, 91; Morris et al., 96; Anselma, 04]). Some of such approaches tackled the problem of dealing with “periodicity-based” temporal constraints, in which constraints between periodic events depend on the specific periodicity in which events occur. For instance, in [Loganathanaraj & Gimbrone, 95], the authors dealt with “periodicity-dependent” durations (e.g., *During the early morning (5am-7am), going to work by car takes from 20 to 30 minutes; During morning rush hours (7am-9am), going to work by car takes from 30 to 40 minutes*), while in [Terenziani, 97] “periodicity-based” qualitative (e.g., “before” [Allen, 83]) constraints were considered.

At the same time, mostly in the area of Temporal Databases, many authors have realized that, since the use of a periodicity or calendar depends on the cultural, legal, and even business orientation of the users (see e.g., [Soo & Snodgrass, 93]), dealing with *user-defined* periodicities is

an essential task. As a consequence, several approaches (consider, e.g., [Leban, 86; Niezette & Stevenne, 92; Bettini & De Sibi, 99; Ning et al., 02; Terenziani, 03] and the survey in [Tuzhilin & Clifford, 95]) have been devised to model user-defined periodicities (and the related notions of *granularities* and *calendars*).

The goal of this paper is to propose a comprehensive framework dealing with both *quantitative* and *qualitative* “periodicity-based” temporal constraints between periodic events, taking also into account *user-defined* periodicities.

2 Representing periodicity-based constraints

In our approach, a (binary) periodicity-based constraint is modelled by a triple

$$\langle Ev, Per, Con \rangle$$

where *Ev* denotes a pair of events, *Per* a user-defined periodicity, and *Con* the temporal constraints¹.

The (intuitive) semantics of periodicity-based constraints $\langle \langle ev_1, ev_2 \rangle, P, C \rangle$ in our approach is the following.

*ev*₁ and *ev*₂ are two periodic events. For *each* occurrence (instance) *p*_{*i*} of the periodicity *P*,

- (i) there is exactly an instance *e1_i* of *ev*₁ taking place in *p_i*;
- (ii) there is exactly an instance *e2_i* of *ev*₂ taking place in *p_i*;
- (iii) *C(e1_i, e2_i)* holds.

Since the main goal of our work is that of devising a comprehensive approach covering all the different issues addressed in the introduction (and the interactions between them), in the current version we have chosen to rely as much as possible on contributions described in literature, both as concern the formalism to define periodicities, and the formalism coping with temporal constraints. In both cases, we have selected widely known and used AI approaches: the *collection* formalism [Leban et al., 86] and

¹ Durations are a degenerate case, in which the *Ev* component only consists of one event, and the *Con* component of the distance between its endpoints.

STP (Simple Temporal Problem [Dechter et al., 91]).

The collection formalism allows one to represent user-defined periodicities in a symbolic, user-friendly and incremental way. Leban et al. take *calendars* (e.g., seconds, days, years) as basic elements, and provide two classes of operators: *dicing* and *slicing*. The *dicing* operators provide a means to further divide each interval in a periodicity according to another periodicity. For example, *Days:during:Months* breaks up months into days. *Slicing* operators provide a way of selecting intervals from collections. For instance, $[1] \setminus \text{Days:during:Weeks}$ selects the first day in each week. Collection expressions can be arbitrarily built by using a combination of these operators. The semantics of the expressions is informally represented in [Leban et al., 86] as nested sets of convex time intervals. However, as in [Bettini & DeSibi, 99], we do not consider the possibility of nesting sets of intervals, in fact, we choose to rely on a simpler notion of temporal extent of a periodicity.

Definition: Extension function. We define the extension (Ext) of a periodicity P the set of time intervals which constitute the instances of P on the timeline. The function Ext takes in input a periodicity (e.g., Mondays) and gives in output a set of time intervals (e.g., supposing that the basic granularity is days and that day 0 is a Monday, $\text{Ext}(\text{Mondays}) = \{ \dots [-7, -7], [0, 0], [7, 7], \dots \}$). $\square$

We allow users to associate names with expressions in Leban et al.'s formalism, and only user-defined names can be used in the *Per* component of the constraints.

The STP framework, on the other hand, is based on the notion of *bound on difference* constraints. A bound on difference is a linear inequality of the form $d_1 \leq X - Y \leq d_2$ (also strict inequalities are allowed). The variables in a bound on difference may correspond to time points (and extremes of time intervals). Thus, a bound on difference $d_1 \leq X - Y \leq d_2$ has the following temporal interpretation: the temporal distance between the time points (endpoints of time intervals in our case) X and Y is included into the *admissibility range* $[d_1, d_2]$. A temporal knowledge base (KB) is a *conjunction* of bounds of differences constraints (called *STP constraints* in [Dechter et al., 91] and henceforth). In [Meiri, 91; Brusoni et al., 97] it is shown how different types of qualitative and quantitative temporal constraints can be easily mapped to STP constraints. In particular, all the qualitative constraints of the *Continuous Interval Algebra* (i.e., the subset of relations of Allen's Algebra which can be mapped onto conjunctions of constraints between points, excluding inequality [Vila, 94]) can be mapped onto STP constraints, as well as precise and imprecise dates, durations and delays. Given an STP, the strictest admissibility range between each pair of time points (i.e., the *minimal network*) can be obtained using a standard all-to-all shortest path algorithm such as, e.g., Floyd-Warshall's one, which operates in a time cubic in the number of points, is correct and complete for STP constraints, and can be trivially adapted to check consistency [Dechter et al., 91].

In our approach, the *Con* component of each periodicity-

based constraint can thus be expressed by a quadruple of admissibility ranges, representing the minimal and maximal distance between the starting/ending points of ev_1 and ev_2 , i.e., $\langle \text{start/start}, \text{start/end}, \text{end/start}, \text{end/end} \rangle$.

Finally, the events (*Ev* component) can be simply specified using identifiers.

For example, the constraint “On Monday, Tom starts to work after Mary, and finishes between 10 and 30 minutes after her” can be represented in our formalism as follows (given the definition of Mondays in Leban et al.'s language)²:

$\langle \langle \text{Tom_work}, \text{Mary_work} \rangle, \text{Mondays}, \langle (0, +\infty)(-\infty, +\infty), (10, +\infty), [10m, 30m] \rangle \rangle$.

3 Intensional calculus

3.1 Intensional vs extensional calculus

Given a KB of periodicity-based temporal constraints, we aim at providing a constraint-propagation-based form of temporal reasoning to amalgamate them and to check their consistency. As most approaches in AI, our temporal reasoning approach is based on the operations of intersection and composition. Unfortunately, usual approaches in the literature (such as, e.g., Allen [Allen, 83] and Morris et al. [Morris et al., 96]), although synthetically defined and semantically clear, do not apply to our case, since we have to manage two different components (periodicity and STP constraint) and the interactions between them. One possibility would be that of “exploding” each periodicity over a time span of interest (or, in the most general case, on the “least common multiple” of the period of each periodicity in the KB), explicitly generating all the repetitions (instances) of the events over such a time span, imposing all the input temporal constraints on the generated instances, and then using a “standard” constraint-propagation technique to perform temporal reasoning on them. Such an *extensional* approach may work, and in section 4 we devise it in such a way that temporal reasoning is tractable and complete. However, the extensional approach has two main drawbacks:

- (i) in most cases, too many instances of events need to be taken into account (since, e.g., the least common multiple of “week” and “month” is 28-year long);
- (ii) in any case, the output of constraint propagation would not be user-friendly and perspicuous (e.g., no train schedule lists explicitly the departure-arrival times of trains in all the days of a year; see e.g. [Terenziani, 03]).

Therefore, we also aim at defining an *intensional* calculus, which, although not complete, performs part of the inferences providing a perspicuous (i.e., intensional) output to users. Such a calculus is defined in this section, and can be combined with the extensional one in order to grant also

² A more user-friendly interface language can be easily provided, adding some “syntactic sugar” to our basic formalism.

completeness in consistency checking³.

3.2 Intensional calculus: main issues

Our basic idea is that of defining a modular intensional calculus, in which intersection ($\cap$) and composition ($@$) are separately computed on the periodicities and on the constraints, i.e.,

$$\begin{aligned} \langle\langle ev_1, ev_2 \rangle, P_1, C_1 \rangle \cap \langle\langle ev_1, ev_2 \rangle, P_2, C_2 \rangle &= \\ \langle\langle ev_1, ev_2 \rangle, P_1 \cap^P P_2, C_1 \cap^C C_2 \rangle & \\ \langle\langle ev_1, ev_2 \rangle, P_1, C_1 \rangle @ \langle\langle ev_2, ev_3 \rangle, P_2, C_2 \rangle &= \\ \langle\langle ev_1, ev_3 \rangle, P_1 @^P P_2, C_1 @^C C_2 \rangle & \end{aligned}$$

The definition of the operators of intersection and composition between STP constraints are the standard ones in Floyd-Warshall's algorithm, i.e.,

$$\begin{aligned} C_1 \cap^C C_2 &= C_1 \cap C_2, \text{ where } \cap \text{ is the standard intersection between time intervals (admissibility ranges);} \\ C_1 @^C C_2 &= K, \text{ where } K = [a+c, b+d] \text{ for } C_1=[a,b] \text{ and } C_2=[c,d]. \end{aligned}$$

On the other hand, several problems have to be faced in order to provide an intensional definition of intersection and composition between user-defined periodicities. Specifically, operating at the intensional level means taking into account only one “prototypical” repetition, i.e., the repetition in a “typical” common period (i.e., $P_1 \cap^P P_2$ or $P_1 @^P P_2$), represented in an intensional way. This means that:

- (1) a one-to-one correspondence between the instances of P_1 and P_2 must be pointed out;
- (2) the formalism for representing periodicities must be extended to be able to intensionally express the common period;
- (3) an algorithm must be devised to compute the common period on the basis of two periodicities P_1 and P_2 . Notice that the computation must operate at the intensional level, and provide an intensional output.

3.3 One-to-one correspondences between periodicities

Given the points (i) and (ii) in the definition of the semantics of our periodicity-based constraints (see section 2), two constraints $\langle\langle ev_1, ev_2 \rangle, P_1, C_1 \rangle$ and $\langle\langle ev_1, ev_2 \rangle, P_2, C_2 \rangle$ referring to the *same* repeated events ev_1 and ev_2 are consistent only in case the periodicities P_1 and P_2 are such that

- (1) there is a 1 to 1 correspondence between instances of P_1 and instances of P_2 , and

- (2) the corresponding instances must intersect in time.

For instance the periodicities Tue&Wed and Wed&Thu (with the intuitive meaning) satisfy the above constraints (so that one can consistently assert that ev_1 occurs exactly once in each Tue&Wed and exactly once in each Wed&Thu, implying that ev_1 occurs exactly once in the intersection). On the other hand, Tue&Wed and Days do not satisfy constraint (1) (since Days are more “frequent” than Tues&Wed), Tue&Wed and Thu&Fry do not satisfy (2).

Unfortunately, since periodicities are intrinsically cyclic, more than one correspondence (specifically, two) satisfying the above constraints (1) and (2) may hold between pairs of periodicities. For instance, the periodicities $P_1=\text{Tue\&Wed\&Thu\&Fri}$ and $P_2=\text{Fri\&Sat\&Sun\&Mon\&Tue}$ “circularly” intersect twice so that two one-to-one correspondences are possible (one pairing each instance of P_1 with the instance of P_2 starting the preceding Friday, and one pairing it with the following instance).

To disambiguate, we first introduce the notion of Reference Time point (RT), as a point working as an absolute reference for all the periodicity-based constraints in a KB. Any disambiguating rule could work. In the following, we just propose one possibility, as a working example.

Anchoring Rule. Given two periodicities P_1 and P_2 , take as the “anchoring pair” the pair consisting of the first instance of P_1 and P_2 which start after RT; the following pairs of the correspondence are obtained by pairing the second, third, etc. instances of P_1 and P_2 (and analogously for the preceding pairs). $\square$

Thus, one-to-one correspondences between periodicities can be unambiguously defined as follows:

Definitions: We call “anchoring relation” between two periodicities P_1 and P_2 (denoted by $A_{RT}(Ext(P_1), Ext(P_2))$), the one-to-one correspondence (if any) obtained by pairing the intersecting instances of P_1 and P_2 , and applying the anchoring rule in case of ambiguity. We state that there is a **one-to-one correspondence** $Cor^{A_{RT}}(P_1, P_2)$ if and only if there is an anchoring relation A_{RT} between $Ext(P_1)$ and $Ext(P_2)$. $\square$

Property 1. The relation A_{RT} is reflexive, symmetric and transitive. $\square$

3.4 Extending the symbolic formalism for user-defined periodicities

In order to define an algebraic approach (i.e., an approach in which the results of intersection and composition are still constraints in our formalism), it is useful to further extend the formalism to represent user-defined periodicities with two operators: pairwise intersection ($\cap^P$) and restricted pairwise union ($\cup^P$) between periodicities. Since they are used to represent in an intensional way “typical” common periods, the operators must take in input periodicities whose instances are in a one-to-one correspondence, and operate on each pair.

³ In principle, it is possible to devise a “hybrid” approach in which one (i) takes in input the symbolic intensional expressions (ii) convert them into extensions (iii) performs operations on extensions and (iv) re-transforms the extensional results into “significant” output intensional expressions. However, such an approach seems to us computationally expensive and quite problematic in practice. In fact, in general, intensional expressions automatically generated from a set of extensions might be far from being “significant” for users (unless very skilful and complex *learning* techniques are devised and used to reach such a goal). Thus, we believe that, if one wants to give in output “perspicuous” intensional results, the evaluation should be performed directly at the intensional level.

Definition (pairwise intersection). Given two periodicities P_1 and P_2 such that $\text{Cor}^{\text{A}_{\text{RT}}}(\mathbf{P}_1, \mathbf{P}_2)$ holds, $P = (P_1 \sqcap^p P_2)$ is defined as follows (where $p_{1h} \cap p_{2k}$ denotes standard intersection between two time intervals –conceived as sets of time points):

$$\text{Ext}(P) = \{(p_{1h} \cap p_{2k}) \setminus \langle p_{1h}, p_{2k} \rangle \in \text{A}_{\text{RT}}(\text{Ext}(P_1), \text{Ext}(P_2))\}. \quad \square$$

Property 2. If $P = (P_1 \cap^p P_2)$, then $\text{Cor}^{\text{A}_{\text{RT}}}(P, P_1)$ and $\text{Cor}^{\text{A}_{\text{RT}}}(P, P_2)$ hold. $\square$

Definition (pairwise restricted union). Given two periodicities P_1 and P_2 such that $\text{Cor}^{\text{A}_{\text{RT}}}(\mathbf{P}_1, \mathbf{P}_2)$ holds, and such that $\forall \langle p_{1h}, p_{2k} \rangle \in \text{A}_{\text{RT}}(\text{Ext}(P_1), \text{Ext}(P_2))$ p_{1h} and p_{2k} intersect or meet in time, then $P = (P_1 \sqcup^p P_2)$ is defined as follows (where $p_{1h} \cup p_{2k}$ denotes standard union between two time intervals –conceived as sets of time points):

$$\text{Ext}(P) = \{(p_{1h} \cup p_{2k}) \setminus \langle p_{1h}, p_{2k} \rangle \in \text{A}_{\text{RT}}(\text{Ext}(P_1), \text{Ext}(P_2))\}. \quad \square$$

Notice that, given the second condition in the definition, $\sqcup^p$ only generates convex time intervals and the restricted union of non-intersecting intervals is empty⁴.

Property 3. If $P = (P_1 \sqcup^p P_2)$, then $\text{Cor}^{\text{A}_{\text{RT}}}(P, P_1)$ and $\text{Cor}^{\text{A}_{\text{RT}}}(P, P_2)$ hold. $\square$

In the following, we distinguish between *base periodicities* and *composite periodicities*. With *base periodicities* we will intend Leban's *calendars* and those periodicities obtained by the application of Leban et al.'s language [Leban et al., 86] only. With the term *composite periodicity* we intend periodicities defined using at least one of the operators (i.e., $\sqcap^p$ and $\sqcup^p$).

3.5 Relations between periodicities

In order to provide "perspicuous" results, the intensional operations of intersection and composition cannot just give in output a string concatenation of input periodicities and operators. At least two types of simplifications can be (and need to be) performed:

- (i) **redundancy elimination**; e.g., the output of the intersection of "Working-Days" (i.e., days from Monday to Friday) and "Mondays" should be just "Mondays" and not "Working-Days $\sqcap^p$ Mondays";
- (ii) **empty periodicity detection**; for instance, the output of the intersection of "Mondays" and "Wednesdays" should be empty and not "Mondays $\sqcap^p$ Wednesdays".

In the next section we present an intensional definition that performs these simplifications. Since periodicities are user-defined, it is not possible to define a priori all the intersections and compositions between all pairs of periodicities.

⁴ The use of restricted union allows us to operate only on convex time intervals, which makes the extensional part of our approach tractable. Notice also that we use union in order to intensionally define composition (see section 3.6), and composition gives an inconsistency whenever the periodicities to be composed do not intersect in time.

Therefore, we pointed out a set of five possible relations between two user-defined periodicities which are exhaustive and mutually exclusive, and we defined intersection and composition on these bases.

The relations are $\sqsubseteq^p, \supset^p, i^p, ni^p, \#^p$:

- (1) $\mathbf{P}_1 \sqsubseteq^p \mathbf{P}_2 \Leftrightarrow \text{Cor}^{\text{A}_{\text{RT}}}(\mathbf{P}_1, \mathbf{P}_2) \wedge \forall \langle p_{1h}, p_{2k} \rangle \in \text{A}_{\text{RT}}(\text{Ext}(P_1), \text{Ext}(P_2)) \ p_{1h} \sqsubseteq p_{2k}$
- (2) $\mathbf{P}_1 \supset^p \mathbf{P}_2 \Leftrightarrow \text{NOT } \mathbf{P}_1 \sqsubseteq^p \mathbf{P}_2 \wedge \text{Cor}^{\text{A}_{\text{RT}}}(\mathbf{P}_1, \mathbf{P}_2) \wedge \forall \langle p_{1h}, p_{2k} \rangle \in \text{A}_{\text{RT}}(\text{Ext}(P_1), \text{Ext}(P_2)) \ p_{1h} \supseteq p_{2k}$
- (3) $\mathbf{P}_1 i^p \mathbf{P}_2 \Leftrightarrow \text{NOT } \mathbf{P}_1 \sqsubseteq^p \mathbf{P}_2 \wedge \text{NOT } \mathbf{P}_1 \supset^p \mathbf{P}_2 \wedge \text{Cor}^{\text{A}_{\text{RT}}}(\mathbf{P}_1, \mathbf{P}_2) \wedge \forall \langle p_{1h}, p_{2k} \rangle \in \text{A}_{\text{RT}}(\text{Ext}(P_1), \text{Ext}(P_2)) \ p_{1h} \cap p_{2k} \neq \emptyset$
- (4) $\mathbf{P}_1 ni^p \mathbf{P}_2 \Leftrightarrow \text{Cor}^{\text{A}_{\text{RT}}}(\mathbf{P}_1, \mathbf{P}_2) \wedge \exists \langle p_{1h}, p_{2k} \rangle \in \text{A}_{\text{RT}}(\text{Ext}(P_1), \text{Ext}(P_2)) \ p_{1h} \cap p_{2k} = \emptyset$
- (5) $\mathbf{P}_1 \#^p \mathbf{P}_2 \Leftrightarrow \text{NOT } \text{Cor}^{\text{A}_{\text{RT}}}(\mathbf{P}_1, \mathbf{P}_2)$

(where $p_{1h} \supseteq p_{2k}$, $p_{1h} \sqsubseteq p_{2k}$ and $p_{1h} \cap p_{2k}$ respectively denote standard equality, inclusions and intersection between two time intervals).

Given two user-defined *base* periodicities, it is possible to devise different sets of rules in order to compute the relation holding between them on the basis of their definition. For example, from any Leban's definition of the form $P_1 = n / P_2$:during: P_3 , we have $P_1 \sqsubseteq^p P_3$, $P_1 \#^p P_2$, $P_2 \#^p P_3$. Unfortunately, this set of rules is not complete; whenever the relation cannot be inferred via the rules, it can be asked to the user (or computed by generating and checking extensions).

3.6 Intensional definition of intersection and composition about basic periodicities

Periodicity intersection and composition are defined as shown in Table 1. Each row in Table 1 must be interpreted as a conditional *simplification* rule of the form: "If Relation Then $P_1 \text{ OP}^p P_2 = \text{formula}$ ", where OP^p is $\cap^p$ or $@^p$.

Periodicity Intersection. Intersection applies to pairs of constraints of the form $\langle \langle ev_1, ev_2 \rangle, P_1, C_1 \rangle$ and $\langle \langle ev_1, ev_2 \rangle, P_2, C_2 \rangle$. For example, the first row states that if $P_1 \sqsubseteq^p P_2$, then $P_1 \cap^p P_2 = P_1$. The last row states that, if there is no one-to-one correspondence between P_1 and P_2 , or the corresponding instances do not intersect, then the intersection is empty, so that the result of $\langle \langle ev_1, ev_2 \rangle, P_1, C_1 \rangle \cap \langle \langle ev_1, ev_2 \rangle, P_2, C_2 \rangle$ is an inconsistency.

Periodicity Composition. Composition applies to pairs of constraints of the form $\langle \langle ev_1, ev_2 \rangle, P_1, C_1 \rangle$ and $\langle \langle ev_2, ev_3 \rangle, P_2, C_2 \rangle$. Concerning the periodicities P_1 and P_2 , notice that the semantics of our constraints imposes that the "common" event ev_2 must occur both exactly once in each P_1 and exactly once in each P_2 . As Table 1 reports, this is

Relation between $\mathbf{P}_1$ and $\mathbf{P}_2$	$\mathbf{P}_1 \cap^p \mathbf{P}_2$	$\mathbf{P}_1 @^p \mathbf{P}_2$
$\mathbf{P}_1 \sqsubseteq^p \mathbf{P}_2$	$\mathbf{P}_1$	$\mathbf{P}_2$
$\mathbf{P}_1 \supset^p \mathbf{P}_2$	$\mathbf{P}_2$	$\mathbf{P}_1$
$\mathbf{P}_1 i^p \mathbf{P}_2$	$\mathbf{P}_1 \sqcap^p \mathbf{P}_2$	$\mathbf{P}_1 \sqcup^p \mathbf{P}_2$
$\mathbf{P}_1 ni^p \mathbf{P}_2$	$\emptyset$	$\emptyset$
$\mathbf{P}_1 \#^p \mathbf{P}_2$	$\emptyset$	$\emptyset$

Table 1. Intersection and composition between periodicities.

possible in case the relation between P_1 and P_2 is one of $P_1 \supset^p P_2$, $P_1 \subseteq^p P_2$ and $P_1 \dot{\supset}^p P_2$, while an inconsistency must be reported in the cases $P_1 \dot{\cap}^p P_2$ and $P_1 \#^p P_2$.

Property 4. Correctness and simplification completeness.

Our definitions of intersection and composition (Table 1) are correct; moreover, they are complete as regards emptiness detection and redundancy elimination. $\square$

Proof (hint): correctness can be proved by showing that all simplifications are correct, i.e., (i) for each rule **IF $P_1 \mathbf{R} P_2$ THEN $\text{Expr1} \rightarrow \text{Expr2}$** in Table 1, then $\text{Ext}(\text{Expr1}) = \text{Ext}(\text{Expr2})$, and (ii) we output an empty value just in case intersection/composition is not possible, given the semantics of the periodicity-based constraints.

Simplification completeness is proven by showing that, given any two base periodicities P_1 and P_2 and an operation OP^p ($@^p$ or $\cap^p$) between them, (i) whenever $\text{Ext}(P_1 \text{OP}^p P_2) \subseteq \text{Ext}(P_2)$, there is a rule such that the intensional expression " $P_1 \text{OP}^p P_2$ " is transformed into " P_2 " (or vice versa, if $\text{Ext}(P_1 \text{OP}^p P_2) \supseteq \text{Ext}(P_2)$); (ii) whenever intersection/composition are not possible, given the semantics of the periodicity-based constraints, there is a rule that reports an empty value. $\square$

3.7 Intensional definition of intersection and composition about composite periodicities

In the following, we hypothesize that composite periodicities are put in normal form, as below:

$(P_{11} \dot{\cup}^p P_{12} \dot{\cup}^p \dots \dot{\cup}^p P_{1st}) \dot{\cap}^p \dots \dot{\cap}^p (P_{h1} \dot{\cup}^p P_{h2} \dot{\cup}^p \dots \dot{\cup}^p P_{hsh})$ where each P_{ij} is a basic periodicity. We also hypothesize that all input composite periodicities (if any) have already been simplified (this can be done in a pre-compilation step). The following property holds.

Property 5. Pairwise restricted union is distributive wrt pairwise intersection. $\square$

Given these premises and given the definitions in Table 1, temporal reasoning on composite periodicities requires to calculate:

- (i) the intersection $\dot{\cap}^p$ of normal forms;
- (ii) the restricted union $\dot{\cup}^p$ of normal forms (to compute $\cap$ and $@$ respectively: see section 3.2).

Basically, operations (i) and (ii) consist of a set of simplification rules to detect emptiness and to eliminate redundancies. Since periodicities are user-defined, operating directly on composite periodicities is a hardly feasible approach. In particular, it is difficult to find rules to determine which one of the five relations holds between two composite periodicities (analogous to the rule in section 3.5). Therefore, the only practical way of operating is that of devising a compositional and modular calculus, in which operations between composite periodicities are decomposed by considering pairwise the basic periodicities composing them.

Note 1. Unfortunately, since certain simplifications can only be captured considering composite periodicities as a whole, such a compositional approach cannot be simplification-

complete (see Property 6).

Intersection of normal forms.

$$(\alpha) ((P_{11} \dot{\cup}^p \dots \dot{\cup}^p P_{1st}) \dot{\cap}^p \dots \dot{\cap}^p (P_{g1} \dot{\cup}^p \dots \dot{\cup}^p P_{gst})) \dot{\cap}^p ((P'_{11} \dot{\cup}^p \dots \dot{\cup}^p P'_{1st'}) \dot{\cap}^p \dots \dot{\cap}^p (P'_{h1} \dot{\cup}^p \dots \dot{\cup}^p P'_{hth})).$$

Emptiness detection. If there are two basic periodicities P_{il} and P'_{jk} such that $P_{il} \#^p P'_{jk}$, then the intersection of the two normal forms is undefined.

If there exist two unions of basic periodicities $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ and $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ such that $\forall P_{il}$ in $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ and $\forall P'_{jk}$ in $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ $P_{il} \dot{\cap}^p P'_{jk}$, the intersection of the two normal forms is empty.

If no emptiness is detected, the resulting periodicity can be obtained by applying the redundancy elimination rules to the formula obtained by concatenating by $\dot{\cap}^p$ the two input periodicities (i.e., to a formula like (α)).

Redundancy elimination. If there are two unions of basic periodicities $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ and $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ such that $\exists P'_{jk}$ in $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ s.t. $\forall P_{il}$ in $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ $P_{il} \subseteq^p P'_{jk}$, then all the other P'_{jp} with $p \neq k$ can be removed from $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ because they surely are unnecessary for computing the result of the intersection (and vice versa, by exchanging the role of the two union sets).

Moreover, if $\exists P'_{jk}$ in $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ s.t. $\forall P_{il}$ in $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ $P_{il} \dot{\cap}^p P'_{jk}$, P'_{jk} can be removed (and vice versa).

Union of normal forms.

$$(\beta) ((P_{11} \dot{\cup}^p \dots \dot{\cup}^p P_{1st}) \dot{\cap}^p \dots \dot{\cap}^p (P_{g1} \dot{\cup}^p \dots \dot{\cup}^p P_{gst})) \dot{\cup}^p ((P'_{11} \dot{\cup}^p \dots \dot{\cup}^p P'_{1st'}) \dot{\cap}^p \dots \dot{\cap}^p (P'_{h1} \dot{\cup}^p \dots \dot{\cup}^p P'_{hth})).$$

By applying property 5, the composite periodicity above can be rewritten as follows:

$$((P_{11} \dot{\cup}^p \dots \dot{\cup}^p P_{1st}) \dot{\cup}^p (P'_{11} \dot{\cup}^p \dots \dot{\cup}^p P'_{1st'})) \dot{\cap}^p \dots \dot{\cap}^p ((P_{g1} \dot{\cup}^p \dots \dot{\cup}^p P_{gst}) \dot{\cup}^p (P'_{h1} \dot{\cup}^p \dots \dot{\cup}^p P'_{hth}))$$

which reduces to the intersection of unions in the form $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi}) \dot{\cup}^p (P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$.

Emptiness detection. If there exist two basic periodicities P_{il} and P'_{jk} such that $P_{il} \#^p P'_{jk}$, then the union of the two normal forms is undefined.

Moreover, let us consider the formula obtained after the application of the distributive property. If there is a union of unions $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi}) \dot{\cup}^p (P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ which is empty, the overall intersection is empty (from our definition, restricted union may introduce emptiness). Working on the corresponding basic periodicities, this translates to the following rule: if $\forall P_{il}$ in $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ and $\forall P'_{jk}$ in $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ $P_{il} \dot{\cap}^p P'_{jk}$, then $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi}) \dot{\cup}^p (P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ (and the whole union of normal forms) is empty.

If no emptiness is detected, the resulting periodicity can be obtained by applying the redundancy elimination rules (to a formula like (β)).

Redundancy elimination. Every union in the form $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi}) \dot{\cup}^p (P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ can be simplified as follows: if $\exists P_{il}$ in $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ and $\exists P'_{jk}$ in $(P'_{j1} \dot{\cup}^p \dots \dot{\cup}^p P'_{jju})$ s.t. $P_{il} \subseteq^p P'_{jk}$, then P_{il} can be removed from $(P_{i1} \dot{\cup}^p \dots \dot{\cup}^p P_{isi})$ (and vice versa).

Moreover, the rules concerning intersection of normal forms can also be applied at this point.

Property 6. Correctness and simplification incompleteness. The simplification rules of empty periodicity detection and redundancy elimination, applicable to composite periodicities, are correct; they are not simplification complete. $\square$ Proof(hint). The correctness proof is similar to the one for basic periodicities; concerning incompleteness, see Note 1.

3.8 Constraint propagation algorithm

Since our overall intensional calculus cannot be simplification-complete (see Property 6), we chose to adopt a low-complexity propagation algorithm. We chose Floyd-Warshall's one, since at least it is complete as regards the *Con* part of our constraints.

1. for each ev_k in periodic events
2. for each ev_i, ev_j in periodic events
3. let PBC_{ij} the constraint between ev_i and ev_j
4. $PBC_{ij} \leftarrow PBC_{ij} \cap (PBC_{ik} @ PBC_{kj})$

The algorithm iteratively applies the operations of intersection and composition a cubic number of times. Thus, given the properties in sections 3.6 and 3.7, the following property trivially holds:

Property 7. Our intensional temporal reasoning approach is correct, but it is not simplification complete. $\square$

Notice also that our intensional approach is non-complete as regards consistency checking. For instance, (i) *every Monday ev_1 starts at least 20 hours before ev_2* , and (ii) *every Monday ev_2 starts at least 10 hours before ev_3* , are jointly inconsistent.

However, such an inconsistency can only be detected by considering the delay constraint together with the periodicities, in an integrated way. A complete extensional algorithm to check consistency is provided in section 4.

4. Extensional calculus

Let LCM be the least common multiple period of the periodicities in the KB. The following algorithm can be used in order to extensionally check the consistency of the KB:

1. generate the extensions of the periodicities relative to the time span LCM representing each time interval by means of its starting and ending point and add the constraints about duration and delay between the time intervals;
2. generate the instances of the events and add the temporal constraints in the KB;
3. add the constraints that the instances are contained in the time intervals of the proper periodicity;
4. propagate the constraints via an all-pairs shortest path algorithm.

Property 8. The extensional calculus is correct, complete and tractable (cubic in the number of the instances of periodicities and periodic events in a LCM period). $\square$

5. Conclusions and comparisons

In this paper, we have described a comprehensive approach dealing with both (i) *qualitative* and (ii) *quantitative* periodicity-based temporal constraints, and considering also (iii) *user-defined* periodicities.

Dealing with these issues and with the interplay between them required to devise a novel approach that integrates and extends the STP framework [Dechter et al., 91] and the Leban's formalism [Leban et al., 86]. In particular, we have: (1) proposed a formalism to represent periodicity-based temporal constraints; (2) extended Leban's formalism to represent composite periodicities; (3) singled out five relations between periodicities and used them to (4) define the operations of intersection and composition among both basic and composite periodicity-based temporal constraints; (5) described an *intensional* approach, which is correct and provides perspicuous (intensional) output, but is not complete, and (6) an *extensional* approach, which can be used to check consistency, and which is correct and complete.

Several AI approaches have dealt with temporal constraints between periodic events. However, to the best of our knowledge, none of them proposed a comprehensive approach covering all the issues (i)-(iii) above. For example, in [Ligozat, 91; Morris et al., 96] only periodicity-independent (e.g., holding "*always*") qualitative constraints between repeated events have been considered. In [Bettini et al., 02], temporal constraints expressed at multiple user-defined granularities between non-repeated events have been tackled. Only periodicity-based quantitative constraints have been considered in [Loganathanaraj & Gimbrone, 96] and only periodicity-based duration constraints in [Terenziani, 97]. Notice also that none of the above cited approaches has provided an intensional calculus about user-defined periodicities, which, on the other hand, have been devised by some algebraic approaches to Temporal DB (consider, e.g., [Niezette & Stevenne, 92; Terenziani, 03]).

References

- [Allen, 83] J.F. Allen. "Maintaining Knowledge about Temporal Intervals", *Comm. ACM*, 26(11):832-843, 1983.
- [Anselma, 04] L. Anselma, "Recursive Representation of Periodicity and Temporal Reasoning", *Proc. TIME 2004*, IEEE Society Press, pp. 52-59, 2004.
- [Bettini & DeSibi, 99] C. Bettini, R. De Sibi, *Symbolic Representation of User-defined Time Granularities*, *Proc. TIME'99*, IEEE Computer Society, pp. 17-28, 1999.
- [Bettini et al., 02] C. Bettini, S. Jajodia, and X. Wang, Solving multi-granularity constraints networks, *Artificial Intelligence*, 140(1-2):107-152, 2002.
- [Brusoni et al., 97] V. Brusoni, L. Console, B. Pernici, and P. Terenziani. "Later: Managing Temporal Information Efficiently". *IEEE Expert*, 12(4):56-64, July/August 1997.

- [Dechter et al., 91] R. Dechter, I. Meiri, J. Pearl, "Temporal Constraint Networks", *Artificial Intelligence*, 49:61-95, 1991.
- [Ladkin, 86] P. Ladkin, "Time Representation: A Taxonomy of Interval Relations", in Proc. AAAI'86, Philadelphia, PA, pp. 360-366, 1986.
- [Leban et al., 86] B. Leban, D.D. McDonald, D.R. Forster, A representation for collections of temporal intervals, in Proc. AAAI'86, pp. 367-371, 1986.
- [Ligozat, 91] G. Ligozat, "On Generalized Interval Calculi", in Proc. AAAI'91, pp. 234-240, 1991.
- [Loganathanaraj & Gimbrone, 1995] R. Loganathanaraj, and S. Gimbrone, "Representation of, and Reasoning with, Near-Periodic Recurrent Events", In *9th IJCAI Workshop on Spatial and Temporal Reasoning*, 1995.
- [Meiri, 91] I. Meiri, "Combining Qualitative and Quantitative Constraints in Temporal Reasoning", in Proc. AAAI'91, pp. 260-267, 1991.
- [Morris et al., 96] R.A. Morris, W.D. Shoaff, and L. Khatib, "Domain Independent Temporal Reasoning with Recurring Events", *Computational Intelligence*, 12(3):450-477, 1996.
- [Niezette & Stevenne, 92] M. Niezette, and J.-M. Stevenne, "An Efficient Symbolic Representation of Periodic Time", In *Proceedings 1st International Conference Information and Knowledge Management*, Baltimore, Maryland, 1992.
- [Ning et al., 02] P. Ning, X.S. Wang, and S. Jajodia, An algebraic representation of calendars, *Annals of Mathematics and Artificial Intelligence*, 36(1-2):5-38, 2002.
- [Soo & Snodgrass, 93] M. Soo, R. Snodgrass, Multiple Calendar Support for Conventional Database Management Systems, Proc. ITDB'93, 1993.
- [Terenziani, 97] P. Terenziani, "Integrating calendar-dates and qualitative temporal constraints in the treatment of periodic events", *IEEE TKDE* 9(5):763-783, 1997.
- [Terenziani, 03] P. Terenziani, "Symbolic User-defined Periodicity in Temporal Relational Databases", *IEEE TKDE*, 15(2):489-509, March/April 2003.
- [Tuzhilin & Clifford, 95] A. Tuzhilin and J. Clifford, "On Periodicity in Temporal Databases", *Information Systems*, 20(8):619-639, 1995.
- [Vila, 94] L. Vila. "A Survey on Temporal Reasoning in Artificial Intelligence", *AI Communications* 7(1):4-28, 1994.