Qualitative and Quantitative Temporal Constraints About Numerically Quantified Periodic Events

Paolo Terenziani
Dipartimento di Informativa, Universita' di Torino
Corso Svizzera 185, 10149 Torino, Italy
Phone: +39 11 7429244; E-Mail: terenz@di.unito.it

Abstract

The paper describes an integrated temporal formalism which deals with (i) quantitative information about the frame of time and the user-defined calendar-dates (periods) in which periodic events are located, (ii) (possibly multiple) numeric quantifiers indicating the number of repetitions of events and (iii) qualitative relations between periodic events. The paper defines the operations of intersection and composition of temporal specifications in the given formalism, which are used in order to perform temporal reasoning. An algorithm supporting specialised forms of reasoning about the number of repetitions of events is also described. Moreover, the paper introduces an expressive query language for extracting different types of temporal constraints from a knowledge base of temporal specifications in the formalism, sketching the reasoning algorithms needed to answer the queries.

1. Introduction

The interest towards the treatment of periodic (*repeated* [5,11]) events is rapidly increasing in the scientific community. In particular, periodic events are widely studied in temporal databases (TDB) and in Artificial Intelligence (AI). In fact, periodic events are involved in many "intelligent" activities, such as planning, scheduling, workflow analysis, office automation.

A main problem in the treatment of periodic events is that different types of temporal information have to be taken into account. For example, in TDB, most attention is devoted to the treatment of calendar-dates (see, e.g., [2, 3, 4, 6, 14, 15, 16]), which represent the *period* upon which periodic events occur (e.g., "Every first Tuesday of the month" in Ex.1).

Ex.1 "<u>Every first Tuesday of the month</u> the class III_A has a test of French"

In particular, since different calendric systems are used for specifying the *periods* (depending e.g., from cultural and social factors [16]), many approaches provided facilities to deal with user-defined calendric definitions (see, e.g., [3, 9, 15, 16]).

On the other hand, many approaches in AI focused on the treatment of qualitative relations between *periodic events* (see, e.g., [7,8,10,11, 12, 13, 22]), such as the "after" relation in Ex.2

Ex.2 "Each correction of the test of French (for the class III_A) is <u>after</u> the test"

In particular, following Allen's approach to the treatment of qualitative relations [1] (between non-periodic events), also some approaches to periodic events are algebraic approaches, in which (i) a specialised formalism is devised to represent temporal information about periodic events and (ii) the operations of intersection and composition (which must be closed with respect to the representation formalism) are defined; path-consistency algorithms repeatedly applying intersection and composition are then used in order to perform temporal reasoning (consider, e.g., [11, 12, 13, 22]).

However, none of the previous approaches took into account the fact that, in general, qualitative relations are not "absolute", but only holds in a specific *frame time* (e.g., "between 11-9-95 and 14-6-96" in Ex.3) and in a given *period* (e.g., "first Tuesday of the month", in Ex.3), so that one could consistently assert, e.g., Ex.3 and Ex.3':

Ex.3 "<u>Between 11-9-95 and 14-6-96</u>, <u>each first Tuesday of the month</u> the class III_A had a test of French <u>before</u> an hour of Mathematics"

Ex.3' "<u>Between 11-9-95 and 14-6-96</u>, <u>each last Tuesday of the month</u> the class III_A had a test of French <u>after</u> an hour of Mathematics"

The goal of our work is that of providing an Allen's-like algebraic approach dealing with these and others (see below) phenomena arising when dealing with periodic events. In [17,20], we proposed an approach dealing with frame times, periods and qualitative relations between periodic events. In this paper we address the extensions needed in order to face the following 3 tasks:

(1) Dealing with numeric quantifiers stating the number of repetitions of events. Numeric quantifiers are pervasive

in activities such as planning, scheduling, work flow analysis, manufacturing, office automation. In this paper we deal with a wide range of uses of numeric quantifiers: (i) to state the number of repetitions of events in a given period (and at a given frame time), as, e.g., in Ex.4, (ii) to state the cardinality of the collections of instances of two given periodic events which are in a given qualitative relations, such as in "to open the strong-room, turn the key three times to the right and then two times to the left", or in Ex.5. Moreover, (iii) we also allow the use of both types of constraints in the very same temporal specifications, see, e.g., Ex.6

Ex.4 "Between 11-9-95 and 14-6-96, the class III_A had an hour of Mathematics two times each Monday"

Ex.5 "Between 11-9-95 and 14-6-96, each Tuesday the class III_A had an hour of Mathematics before two hours of Physics"

Ex.6 "Between 11-9-95 and 14-6-96, three times each week the class III_A had two hours of Physics before two hours of Chemistry"

(2) Supporting specialised forms of reasoning about the number of repetitions of periodic events in periods.

(3) Answering different types of queries. Despite the fact that querying is the most common way of interacting with a temporal manager, the problem of providing a rich query language and of supporting the reasoning procedures needed in order to answer queries has often been only marginally considered in the AI literature.

The paper is organised as follows. In section 2, we introduce our formalism dealing with periods, numeric quantifiers and qualitative relations between periodic events and we define the operations of intersection and composition used by a path-consistency algorithm to perform temporal reasoning. Section 3 describes an algorithm for reasoning about the number of repetitions of events. Section 4 deal with different types of temporal queries, and sketches the reasoning algorithms to deal with them. Section 5 proposes comparisons and conclusions.

In this paper, we describe the *semantics* of our high-level language in an informal way. In [18] we defined a logic for dealing with the semantics of temporal constraints about periodic events. In [19] we applied such a logic to state in a formal (logical) way the meaning of the temporal specifications of the high-level formalism introduced in this paper, and to prove the corretness of our algebraic operations (see Property 1 in section 2.2.). These issues are not dealt with in this paper, for the sake of brevity.

2. Temporal constraints about numerically quantified periodic events

2.1 The temporal formalism

Our "high-level" temporal formalism has been defined in such a way that temporal constraints such as those discussed in the introduction can be expressed, and path-consistency on a set of temporal specifications in our formalism can be computed in polynomial time. Our formalism provides three basic types of temporal specifications, which can be expressed according to the simplified syntax shown below ("{}" indicates optionality; the syntax in this paper has been simplified, to the sake of clarity and brevity. We also provide "syntactic sugar" to help the user; e.g., to avoid that s/he has to specify many times the same frame-time -or period- for different specifications).

<TEMPORAL_SPECIFICATION> ::=
<LOC> | <QUAL> | <LOC_QUAL>
<LOC> ::= <Frame> <Periodic_Event> {<Num>} EACH <Period_Name>
<QUAL>::=<Frame> EACH {<Num>} <Periodic_Event> <Qual_Rel>{<Num>}<Periodic_Event>
<LOC_QUAL>::= <Frame> {<Num>} EACH <Period_Name> {<Num>} <Periodic_Event> <Qual_Rel> {<Num>} <Periodic_Event>

<Frame> indicates a frame of time (time interval) ranging from a starting point to an ending point, and is represented by a pair of dates (e.g., [11-9-95,14-6-96]).

<Num> is a numeric quantifier representing the *exact* number of repetitions of the events (in the form "n-times"). It can be omitted; in such a case, it assumes the default value "exactly once".

<Periodic_Event> is the representation of an event repeated in time. The algebraic approach we propose is independent of the representation of periodic events (for the sake of clarity, in the following we simply associate an identifier to each periodic event e.g., "correction-test-French-III-A" -see [18] for an alternative representation in first order logic).

<Qual_Rel> is a qualitative relation between the temporal extent of two periodic events, expressed using any relation in Allen's Interval Algebra [1].

<Period_Name> is a user-defined identifier of a period (*calendric definition*; e.g., "1st-Tuesdays-of-Months*"). In our approach, the definitions of the periods must be provided by the user using a slight adaptation of Leban's language [9], which we discussed in [20].

• The **specifications** of type **<LOC>** allow one to deal with the quantitative temporal constraints; they relate a periodic event to the period and frame time in which it happens. For example, (given a definition of the Period_Name "Mondays*" in Leban's language), the temporal constraints in Ex.4 can be represented as shown in (S1), with the meaning that, for each week strictly contained in the frame time "between 11-9-95 and 14-6-96", there are exactly two instances of an hour of lesson of Mathematics of the class III_A (identifier *H_Math_III_A**) contained in it (in this paper, we choose to associate an event to each hour of lesson of a given class).

(S1) [11-9-95,14-6-96] H_Math_III_A* 2-times EACH Mondays*

• **<QUAL> specifications** deal with "period-independent" qualitative temporal relations between periodic events (in a given frame time). They can be used to express "absolute" (independent of the period) temporal constraints in a given frame time. E.g., the temporal constraint in Ex.2 can be represented in our high-level language as shown in (S2), with the meaning that, in the frame time (-∞,+∞), there is a bijective 1:1 relation (called *correlation* [12,17,20]) between instances of correction_test_French_III_A* and of test_French_III_A* such that the temporal relation AFTER holds between each correlated pair.

(S2) (-∞,+∞) EACH correction_test_French_III_A* (AFTER) test_French_III_A*

• **<LOC_QUAL> specifications** deal with "period-dependent" qualitative relations between periodic events, i.e., qualitative relations holding at specific periods of time (i.e., with mixed qualitative and quantitative temporal constraints). E.g., the temporal constraints in Ex.6 can be represented in our formalism as shown in (S3).

(S3) [11-9-95,14-6-96] 3-times EACH Weeks* 2-times H_Phys_III_A* (BEFORE) 2-times H_Chem_III_A*

The meaning of (S3) is that there is a bijective relation (correlation) holding between collections of two instances of H_Phys_III_A* and collections of two instances of H_Chem_III_A* such that, for each instance of Weeks* during [11-9-95,14-6-96] there are exactly three correlated pairs of a collection of two instances of H_Phys_III_A* and a collection of two instances of H_Chem_III_A*, and the temporal relation between the temporal extent of each correlated pair is AFTER. We define the temporal extent of a collection $C=\{e_1, ..., e_n\}$ of n instances of an event as the minimal convex time interval covering the temporal extents (time intervals) of $e_1, ..., e_n$ [18].

2.2 Temporal reasoning

As in many AI approaches to qualitative temporal constraints (see, e.g., the survey in [21]), in our proposal temporal reasoning is performed by the algebraic operations of inversion (which is not discussed here, for the sake of brevity), intersection and composition. However, we also consider periods, frame times and numeric quantifiers. Thus, we had to face new problems when defining intersection and composition (see [17,20] for a detailed discussion; for the sake of brevity, in this section we only consider temporal specifications of type <LOC-QUAL>; the treatment of the other types of specification is simpler, and can be found in [19]).

First of all, no intersection or composition must be performed in the non-intersecting parts of the frame times. Moreover, the result of intersection and composition of two specifications S' and S" in the common part of the frame times depends on the relation holding between the periods in S' and in S" and on the quantifiers. Consider, for instance, the intersection of (S4) and (S5). The AFTER relation in (S4) is not possible between 11-9-95 and 14-6-96, and must be ruled out. More specifically, the intersection of (S4) and (S5) gives as result the specifications (S6), (S7), and (S8). However, in case we substitute the period Mondays* instead of Tuesdays* in (S5), the intersection would provide as result the two input specifications unchanged. Also quantifiers must be checked; e.g., in case we substitute EACH Tuesdays* 2-times test_French_III_A* instead of EACH Tuesdays* test_French_III_A* in (S5), we have an inconsistency. Thus, we cannot propose a compact definition of intersection and composition such as in [1,11,12,13]: we need a definition by cases based on the relation holding between the periods in the specifications and on the quantifiers.

(S4) [11-9-95,31-7-96] EACH 1st-Tuesdays-of-Months* test_French_III_A* (BEFORE,AFTER) H_Math_III_A*

(S5) [11-9-95,14-6-96] EACH Tuesdays* test_French_III_A* (BEFORE,MEETS) H_Math_III_A*

(S6) [11-9-95,14-6-96] EACH 1st-Tuesdays-of-Months* test_French_III_A* (BEFORE) H_Math_III_A*

(S7) [11-9-95,14-6-96] EACH Tuesdays* test_French_III_A* (BEFORE,MEETS) H_Math_III_A*

(S8) (14-6-96,31-7-96] EACH 1st-Tuesdays-of-Months* test_French_III_A* (BEFORE,AFTER) H_Math_III_A*

2.2.1 Relations between periods. In [17,20] we distinguished among six basic relations between periods, plus their inverses (indicated by $^{-1}$; the inverse of $=^T$ is $=^T$; the inverse of # is #), which are exhaustive and mutually exclusive. In the following, we sketch the six relations (C1* and C2* indicates two periods):

• C1* $=^T$ C2* (read as: C1* and C2* are temporally equal) iff there is a bijection between instances of C1* and instances of C2*, and Allen's relation EQUAL holds between each pair of corresponding instances.

• C1* $\prec$ C2* (C1* is more specific than C2*) iff for each instance of C1* there is exactly one instance of C2* which properly contains it and, conversely, for each instance of C2* there is exactly one instance of C1* which is properly contained in it (bijection) (e.g., Tuesdays* $\prec$ Weeks*).

• C1* $\in$ C2* (C1* is a restriction of C2*) iff for each instance of C1* there is an instance of C2* which is temporally equal to it, but not vice versa (e.g., Mondays* $\in$ Days*).

• $C1^* \in^{INC} C2^*$ (C1* is an inclusion restriction of C2*) iff for each instance of C1* there is an instance of C2* which properly contains it, but not vice versa (e.g., Christmas* $\in^{INC}$ Months*).

• $C1^* \subset_I C2^*$ (C1* is more frequent than C2* in the frame time I) covers the cases where two assertions such as (i) "ev1* happens exactly once each C1*" and (ii) "ev1* happens exactly once each C2*" are inconsistent in a frame time I (e.g., Days* $\subset_I$ Weeks* in [1-1-91,1-6-91]).

• C1* # C2* (C1* and C2* are temporally incomparable) iff none of the above relations (or their inverses) hold between C1* and C2* (e.g., Mondays* # Tuesdays*).

In [20] we widely debated the rationale beyond such a distinction, and described a set of heuristic rules for determining automatically which one of the 10 (the 6 above plus 4 inverses) relations holds between two user-defined periods specified using Leban's formalism [9]. Such a set of heuristic rules is not complete, but it is powerful enough to cover non-exceptional cases (remember that, in our approach, the user is completely free in introducing new period definitions, using Leban's formalism), as we discussed in [20]. In case a relation between two user-defined periods is not discovered by the heuristic rules, such a relation is asked to the user.

2.2.2 Definitions of the operations.

Intersection operates on two <LOC_QUAL> temporal specifications of the form of (S9) and (S10) (I stands for the intersection of the frame times), and checks whether R, the intersection in Allen's Interval Algebra of the qualitative relations R1 and R2, is the empty set (in such a case, an inconsistency is reported) and the consistency of the numeric quantifiers, depending on the relations between C1* and C2*.

(S9) I n2'-times EACH C1* n1'-times a* R1 n3'-times b*
(S10) I n2"-times EACH C2* n1"-times a* R2 n3"-times b*

For instance, in case **C1* $=^T$ C2*** or **C1* $\in$ C2***, the numeric quantifiers in the two specifications must be exactly the same. Otherwise, (S9) and (S10) would state a different number of instantiations of a* and/or b* in the given period, or a different grouping of these instances into collections. For instance, (S3) above and (S11) are inconsistent since they involve different groupings of instances of H_Phys_III_A* and H_Chem_III_A*.

(S11) [11-9-95,14-6-96] 2-times EACH Weeks* 3-times H_Phys_III_A* (BEFORE) 3-times H_Chem_III_A*

On the other hand, in case **C1* $\subset_I$ C2*** the relative frequency of the two periods have to be taken into account. For example, (S12) is consistent with (S13), since they both state that H_Math_III_A* and H_Phys_III_A* occur 7 and 14 times each week respectively (but we have an inconsistency if we substitute 5 to 7 in (S12)).

(S12) [11-9-95,14-6-96] 7-times EACH Weeks* H_Math_III_A* (BEFORE) 2-times H_Phys_III_A*
(S13) [11-9-95,14-6-96] EACH Days* H_Math_III_A* (BEFORE) 2-times H_Phys_III_A*

The definition of intersection of two specifications of the form of (S9) and (S10) above is summarised in Table 1(a); R indicates the intersection between the Interval Algebra relations R1 and R2 and RF(C1*,C2*) indicates the relative frequency of C1* with respect to C2* .

Composition operates on two <LOC_QUAL> specifications of the form of (S14) and (S15) and checks the consistency of the number of repetitions of the common event (b* in (S14)-(S15)).

(S14)I n2'-times EACH C1* n1'-times a* R1 n3'-times b*
(S15) I n2"-times EACH C2* n1"-times b* R2 n3"-times d*

If there is no inconsistency, and the numeric quantifiers in the two specifications are the same, new temporal specifications in our formalism can be inferred, depending on the relations between the periods C1* and C2*. E.g., since Tuesdays* $=^T$ Tuesdays*, the composition of (S7) above and (S16) gives as result (S17)

(S16) [11-9-95,14-6-96] EACH Tuesdays* H_Math_III_A* (BEFORE) 2-times H_Phys_III_A*
(S17) [11-9-95,14-6-96] EACH Tuesdays* test_French_III_A* (BEFORE) 2-times H_Phys_III_A*

On the other hand, the composition of (S7) and (S16') (obtained from (S16) by substituting Weeks* to Tuesdays*), cannot provide any new specification expressible in our formalism (this fact is indicated by "NO NEW SPECIFICATION" in Table 1(b)). In fact, it would not be correct to infer either (S18) or (S19). In fact, (S18) states that H_Phys_III_A* occurred (exactly 2 times) each Tuesday (while from (S16') we only have that it occurred 2 times during the week); (S19) states that test_French_III_A* occurred exactly once each week (while from (S7) we only have that it occurred exactly once each Tuesday, so that it could also occur some other times, e.g., on Wednesday).

(S16') [11-9-95,14-6-96] EACH Weeks* H_Math_III_A* (BEFORE) 2-times H_Phys_III_A*
(S18) [11-9-95,14-6-96] EACH Tuesdays* test_French_III_A* (BEFORE) 2-times H_Phys_III_A*
(S19) [11-9-95,14-6-96] EACH Weeks* test_French_III_A* (BEFORE) 2-times H_Phys_III_A*

The definition of composition of two specifications of the form of (S14) and (S15) is summarised in Table 1(b), where R represents the composition of the qualitative relations R1 and R2 in Allen's Interval Algebra. Relation indicates the relation between the periods C1* and C2*.

Relation	Result
$=^T$	if n1'=n1" and n2'= n2" and n3'=n3" and R≠0 then I n2'-times EACH C1* n1'-times a* R n3'-times b* else INCONSISTENT
$\prec$	if n1'=n1" and n2'= n2" and n3'=n3" and R≠0 then I n2'-times EACH C1* n1'-times a* R n3'-times b*, I n2"-times EACH C2* n1"-times a* R n3"-times b* else if n1'=n1" and n2'< n2" and n3'=n3" and R≠0 then I n2'-times EACH C1* n1'-times a* R n3'-times b*, I n2"-times EACH C2* n1"-times a* R2 n3"-times b* else INCONSISTENT
$\in$	if n1'=n1" and n2'= n2" and n3'=n3" and R≠0 then I n2'-times EACH C1* n1'-times a* R n3'-times b*, I n2"-times EACH C2* n1"-times a* R2 n3"-times b* else INCONSISTENT
$\in^{INC}$	if n1'=n1" and n2'≤ n2" and n3'=n3" and R≠0 then I n2'-times EACH C1* n1'-times a* R n3'-times b*, I n2"-times EACH C2* n1"-times a* R2 n3"-times b* else INCONSISTENT
$\subset_I$	if n1'=n1" and n2"/n2'=RF(C1*,C2*) and n3'=n3" and R≠0 then I n2'-times EACH C1* n1'-times a* R n3'-times b* else INCONSISTENT
#	I n2'-times EACH C1* n1'-times a* R1 n3'-times b*, I n2"-times EACH C2* n1"-times a* R2 n3"-times b*

Table 1(a). Intersection

Relation	Result
$=^T$	if n2'=n2" and n3'=n1" then I n2'-times EACH C1* n1'-times a* R n3"-times d* else INCONSISTENT
$\prec$	if (n3'=n1") AND (n2'≤ n2") then NO NEW SPECIFICATION else INCONSISTENT
$\in$	if n2'=n2" and n3'=n1" then I n2'-times EACH C1* n1'-times a* R n3"-times d* else INCONSISTENT
$\in^{INC}$	if (n3'=n1") AND (n2'≤ n2") then NO NEW SPECIFICATION else INCONSISTENT
$\subset_I$	if (n3'=n1") AND ((n2"/n2') = RF(C1*,C2*)) then NO NEW SPECIFICATION else INCONSISTENT
#	NO NEW SPECIFICATION

Table 1(b). Composition

In [19], we provided a logical formalization of (i) the temporal specifications in our formalism, and (ii) the relations between periods (e.g., $\prec$), and used this first order temporal logic to prove Property 1:

Property 1. *Our operations of intersection and composition are correct and do not lose information.*

Property 1 grants that our operations of intersection and composition can be seen as a *compilation* of a set of logical inferences that could also be performed (in a less efficient way), e.g., by a theorem prover for our logic.

In our approach, temporal reasoning is performed by PCforPE [20], a path-consistency algorithm repeatedly applying intersection and composition. PCforPE extends Allen's algorithm [1] to deal with temporal constraints between periodic events. The time *complexity* of PCforPE is cubic in the number of periodic events. PCforPE can be applied also to the temporal specifications discussed in this paper, using the definitions of intersection and composition in this paper instead of the definitions in [20] (where numeric quantifiers had not been taken into account). Given Property 1, Corollary 1 holds; moreover, we also proved Property 2 [19]:

Corollary 1. *PCforPE is correct and does not lose information*

Property 2. *The algebra defined by the formalism described in section 2.1 and by the operations of intersection and composition above (and of inversion) is a conservative extension of the algebra in [20].*

3 Repetitions-check algorithm

PCforPE, repeatedly applying the operation of intersection (and composition), checks for some types of inconsistencies. However, intersection (and composition) operates on two specifications at a time, while certain inconsistencies on the number of repetitions of an event E can only be detected considering all the specifications concerning E at the same time. For example, (a), (b), and (c) are pairwise consistent (thus no inconsistency is found repeatedly applying intersection), but conjuntively inconsistent: (a) event$_X$ exactly twice each Monday, (b) event$_X$ exactly twice each Tuesday, and (c) event$_X$ exactly three times each week (for the sake of brevity, in the following we operate as if all the specifications had the very same frame time -the extensions to deal with multiple different frame times are obvious). The algorithm to detect these types of inconsistencies is based on E_P_REPETITION_CONSIST(E,P,KB) below, which checks the consistency of the number of the repetitions of the event E in the period P, given a knowledge base KB of specifications in our formalism (henceforth we denote set membership by IN, to distinguish it from the relation $\in$ indicating temporal restriction between periods).

Get_repetitions(E,S) is a function retrieving the number of repetitions of the event E, given the specification S only. Period(S) is a function extracting the period from a specification S. For example, given (S3) above we have Get_repetitions(H_Phys_III_A*,S3) = 6; Period(S3) = Weeks*. Relative_Frequency(P1,P2) is defined by cases:

(1) Relative_Frequency(P1,P2)=1 if the relation between the two periods P1 and P2 is $=^T$ or $\prec$ or $\in$ or $\in^{INC}$ or $\prec^{-1}$ or $\in^{-1}$ or $\in^{INC^{-1}}$ or $\in^{-1}$;

e.g., Relative_Frequency(Days*,Mondays*)=1;

(2) Relative_Frequency(P1,P2) = the relative frequency of P1 and P2 if the relation between P1 and P2 is $\subset_I$ or $\subset_I^{-1}$ (in such a case, it gives the same values as the function RF used in the definitions of intersection and

composition; e.g., Relative_Frequency(Days*,Weeks*) = RF(Days*,Weeks*)=7);
(3) Relative_Frequency(P1,P2) = 0 otherwise.

T_Disjoint(P1,P2) is a predicate which is true if the two periods P1 and P2 are temporally disjoint (e.g., Mondays and Tuesdays); T_Cover(P1,P2) is true if the instances of P1 temporally cover completely the instances of P2 (e.g., T_Cover(Days*,Weeks*) is true.

E_P_REPETITION_CONSIST(E,P,KB)
LOC_SPEC $\leftarrow$ set of all specifications S of type <LOC> or <LOC-QUAL> in KB involving the event E;
EXACT_SPEC $\leftarrow$ set of all specifications S' in LOC_SPEC such that P $=^T$ Period(S') or P $\in$ Period(S') or (P $\subset_I^{-1}$ Period(S') and T_Cover(Period(S'),P));
ATLEAST_SPEC $\leftarrow$ set of all specifications S' in LOC_SPEC such that P $\in^{-1}$ Period(S') or
P $\prec^{-1}$ Period(S');
<u>FOR EACH</u> specification S <u>IN</u> EXACT_SPEC <u>**DO**</u>
Get_repetitions(E,S) * Relative_Frequency(Period(S),P);
Check that the number of repetitions is the same for each S in EXACT_SPEC. <u>**OD**</u>
<u>IF</u> the number of repetitions is the same <u>THEN</u>
<u>**BEGIN**</u>
let X be the number of repetitions;
DISJ_A_S $\leftarrow$ set of maximal sets of specifications S' in ATLEAST_SPEC such that $\forall$ Set <u>IN</u> DISJ_A_S, $\forall$ S1, S2 <u>IN</u> Set, the followig hold:
(Period(S1) # Period(S2) AND
T_Disjoint(Period(S1),Period(S2)))
Y $\leftarrow$ $\mathbf{MAX}_{\text{Set IN DISJ\_A\_S}}$ ($\Sigma_{\text{S IN Set}}$
Get_repetitions(E,S) * Relative_Frequency(Period(S),P);
<u>IF</u> X < Y <u>THEN</u> RETURN("Inconsistent")
<u>ELSE</u> RETURN("OK")
<u>**END**</u>
<u>ELSE</u> RETURN("Inconsistent")

E_P_REPETITION_CONSIST is based on the 10 relations between periods in section 2.2.1 (plus the predicates T_Disjoint and T_Cover). The idea is the following. Given a period P, and the set LOC_SPEC of temporal specifications concerning the periodicity of an event E, all the specifications S' in LOC_SPEC such that P is in the relation $=^T$ or $\in$ or $\subset_I^{-1}$ with Period(S') provide the exact number of repetitions of E in P. More specifically, if P $=^T$ Period(S') or P $\in$ Period(S') the number of repetitions can be directly extracted from S' (function Get_repetitions; in this case, Relative_Frequency(P,Period(S'))=1); if P $\subset_I^{-1}$ Period(S') and T_Covers(P,Period(S')) the number of repetitions is the product of Get_repetitions(E,S) with the relative frequency between P and Period(S') (e.g., P=Weeks*, Period(S)=Days*, Relative_Frequency(Period(S),P)=7). On the other hand, all the specifications S' in LOC_SPEC such that P is in the relation $\in^{-1}$ or $\prec^{-1}$ or $\subset_I^{-1}$ Period(S') and not T_Cover(Period(S'),P) with Period(S') provide a lower limit for the number of repetitions of E in P. For example, let us suppose to have (a) *E 2-times EACH Monday*; (b) *E 2-times EACH Sunday*; (c) *E 3-times EACH Week-end*; and that the period P we consider is *Week*. From (a) and (b) we have that E happened at least 2 times each week, and from (c) we have that E happened at least 3 times each week. However, since Monday and week-end are temporally disjoint, from (a) and (c) we have that E happened at least 5 times each week. In general, one has to consider all the maximal sets of specifications such that all the periods they contain are temporally disjoint, and evaluate (using addition) the number of repetitions for each set. Then, the maximum of these number can be taken as lower bound for the repetitions of the event in P. In the example above the maximal set of specifications containing temporally disjoint sets of periods are {(a),(b)} and {(a),(c)}, and the number of repetitions is 4 for the first set, and 5 for the second, so that 5 can be taken as lower bound. Finally, all the specifications S' in LOC_SPEC such that P is in one of the other relations (e.g., #) with Period(S') do not provide any constraint on the number of repetitions of E in P (e.g., if P=Mondays and S=E 2-times EACH Friday).

Thus, E_P_REPETITION_CONSIST checks that all the specifications S in LOC_SPEC such that P is in the relation $=^T$ or $\in$ or $\subset_I^{-1}$ and T_Covers(P,Period(S')) with Period(S') provide the same number of repetitions for E (let X be such a number), and that the specifications S' in LOC_SPEC such that P is in the relation $\in^{-1}$ or $\prec^{-1}$ or $\subset_I^{-1}$ Period(S') and not T_Cover(Period(S'),P) with Period(S') provide a lower limit for the number of repetitions of E which is not greater than X. The overall check on the consistency of the number of repetitions of events in a KB can be simply performed by repeatedly applying the algorithm E_P_REPETITION_CONSIST above until all events and periods in KB are considered, or an inconsistency is detected, and runs in a time linear in the number of periodic events.

4. Query answering

Given a knowledge base KB in which consistency has been checked and the constraints have been propagated (via the path-consistency algorithm plus the repetitions check described in section 3), it is important to provide users and applications with an expressive query language to interact with the knowledge base. Different types of queries are supported in our approch. To exemplify the queries, we consider that the input knowledge base KB_{inp} consists of the specifications (S1), (S4), (S5), (S16) above. KB_{inp} is consistent and PCforPE infers, among the others, the specifications (S6), (S7), (S8), and (S17)). For the sake of brevity, in the following we do not consider frame times.

Yes/no queries. It is important to provide users with the possibility of *asking whether a given set Q of constrains is consistent with the knowledge base KB.* The consistency of a conjunction Q of specifications can be checked (in cubic time) by adding the constraints Q to KB, and by checking the consistency of KB $\cup$ Q as shown in sections 2.2.2 and 3.
e.g.,Consistent?(KB_{inp}, EACH Tuesdays* test_French_III_A* (BEFORE) 2-times H_Phys_III_A*) --> YES

Period-based queries. Given a period P (e.g., P=Monday), one may ask *which events occurred in P (and in which order).* These queries can be answered by considering that all events E' occurred in a period P' such that P $=^T$ P' or P $\in$ P' or P $\prec^{-1}$ P' or P $\subset_I^{-1}$ P' necessarily occurred also in P (e.g., if P=Mondays* and P'=Days*; P $\in$ P'). In all the other cases, events might have occurred in P (e.g., if P=Mondays* and P'=Weeks* -P $\prec$ P'-, since if E' occurred exactly n-times each week, it may be the case that E' occurred n'-times (n'≤n) each Mondays), but this is not certain given KB. Thus, it is enough to retrieve all the specifications whose period P' is in the relation P $=^T$ P' or P $\in$ P' or P $\prec^{-1}$ P' or P $\subset_I^{-1}$ P' with P, and possibly to omit the qualitative constraints between events from the answer, if they have not been explicitly requested in the query.
e.g., Events_in_Period?(KB_{inp},Tuesdays*) --> {2-times H_Phys_III_A*, H_Math_III_A*, test_French_III_A*}

Event-based queries. Given a periodic event E, one may want to *extract all the constraints concerning E from the knowledge base* (periods, number of repetitions, and, optionally, qualitative relations with other events). This can be simply done by a retrieval operation on the KB.
e.g., When_Event?(KB_{inp},H_Phys_III_A*) -->
{EACH Tuesday* test_French_III_A* (BEFORE) 2-times H_Phys_III_A*, EACH Tuesdays* H_Math_III_A* (BEFORE) 2-times H_Phys_III_A*}

Periodicity+Event based queries. Given an event E and a period P, one may ask *how many times did E occur in P.* The algorithm to anwer this type of queries is analogous to E_P_REPETITION_CONSIST. In particular, if KB contains at least one specification S' concerning E such that P $=^T$ Period(S') or P $\in$ Period(S') or (P $\subset_I^{-1}$ Period(S') and T_Cover(Period(S'),P)), the exact number of repetitions can be extracted from S' as shown in E_P_REPETITION_CONSIST. Otherwise, if KB contains a non-empty set of specification S' concerning E such that P $\in^{-1}$ Period(S') or P $\prec^{-1}$ Period(S') or (P $\subset_I^{-1}$ Period(S') and NOT T_Cover(Period(S'),P)), the lower bound of the number of repetitions can be extracted from the specifications in such a set as shown in E_P_REPETITION_CONSIST.
e.g., How_Many_Times(KB_{inp},H_Math_III_A*,Weeks*) --> ATLEAST 3-times

5. Conclusions and comparisons

In [17,20], we proposed an approach dealing with frame times, periods and qualitative relations between periodic events. In this paper we address the extensions needed in order to (1) deal with numeric quantifiers stating the number of repetitions of periodic events, (2) support specialised forms of reasoning about the number of repetitions of events in periods, and (3) support the possibility of answering different types of queries. In particular, we showed how the distinction between 10 basic relations between periods in [17,20] is essential in order to achieve all the extensions above. The approach described in this paper extends to many respect TDB and AI approaches to periodic events. For example, Morris' algebraic approach [12,13] only deals with quantifiers and qualitative temporal constraints such as those in Ex.2 above. In [13], combinations of existential and universal quantifiers are used in order to specify the mapping relations between two periodic events ev1* and ev2* which are in a given qualitative relation. For instance, the quantifier $[\forall\exists_d! \sqcap (\forall\exists_d!)^i]$ corresponds to "always and only", i.e., to a 1:1 mapping between instances of ev1* and instances of ev2*. Our quantifier EACH roughly corresponds to this meaning, but has a wider application, since it states a 1:1 mapping between *numerically quantified collections* of instances of ev1* and ev2*. A wide range of "fuzzy" quantifiers has been considered, e.g., by Ladkin [7,8], who, on the other hand, did not focus on the development of specialised reasoning and query answering techniques. However, to the best of our knowledge, no current approach about periodic events covers all of the aspects considered in this paper, neither from the point of view of the formalism (e.g., qualitative relations between numerically quantified periodic events and period-dependent qualitative relations -see, e.g., Ex.3,3',4,5,6), nor from the point of view of the specialised algorithms for performing temporal reasoning and answering queries.

Especially in the area of TDB, many approaches stressed the importance of dealing with user-defined periods in application areas such as scheduling, financial trading, workflow analysis. Moreover, in these and other areas (e.g., planning) also numeric quantifiers may be important in order to express precisely the number of repetitions of events. The formalism described in this paper deals with these aspects, and has been designed in such a way that path-consistency on a set of statements in our formalism can be computed in polynomial time. For instance, in [20] we showed that the addition of the AND operator to Leban's language for representing periods (e.g., to define periods such as Tuesdays* AND 1st-Days-of-Months*) makes path-consistency intractable.

We are currently investigating the possibility of extending our framework for dealing with quantifiers such as "only", "sometimes" etc. in [12] for associating events to periods.

The approach in this paper constitutes the basis of TeMP$^+$, a prototype of temporal manager dealing with periodic events which extends TeMP [17,20] to deal with numeric quantifiers and queries.

REFERENCES

[1] J.F. Allen: "Maintaining Knowledge about Temporal Intervals", *Comm ACM* 26(11) 832-843 (1983).

[2] M. Baudinet, J. Chomicki, P. Wolper: "Temporal Databases: Beyond Finite Extensions", *Proc. Int'l Works. on an Infrastructure for Temporal Databases*, June 1993.

[3] R. Chandra, A. Segev: "Managing Temporal Financial Data in an Extensible Database", *Proc. 19th Int'l Conference on Very Large Databases*, 1993.

[4] J. Chomicki, T. Imielinsky: "Finite Representation of Infinite Query Answers", *ACM Transactions on Database Systems 18(2)*, 181-223, 1993.

[5] D. Cukierman, and J. Delgrande, "Characterizing Temporal Repetition", *Proc. TIME-96*, Chittaro, Goodwin, Hamilton and Montanari eds., IEEE Press, pp. 80-87, Los Alamitos, CA, USA, 1996.

[6] F. Kabanza, J.-M. Stevenne, P. Wolper: "Handling Infinite Temporal Data", *Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 392-403 (1990).

[7] P. Ladkin: "Primitive and Units for Time Specification", *Proc. AAAI'86*, 354-359 (1986).

[8] P. Ladkin: "Time Representation: A Taxonomy of Interval Relations", *Proc. AAAI'86*, 360-366 (1986).

[9] B. Leban, D.D. McDonald, D.R. Forster: "A representation for collections of temporal intervals", *Proc. AAAI'86*, 367-371 (1986).

[10] G. Ligozat: "On Generalized Interval Calculi", *Proc. AAAI'91*, 234-240 (1991).

[11] R. Loganantharaj, and S. Gimbrone: "Probabilistic Approach for Representing and Reasoning with Repetitive Events", *Proc. eighth Florida Artificial Intelligence Symp.*, pp. 26-30, 1995.

[12] R.A. Morris, W.D. Shoaff, L. Khatib: "Path Consistency in a Network of Non-convex Intervals", *Proc. IJCAI'93*, 655-660 (1993).

[13] R.A. Morris, W.D. Shoaff, L. Khatib: "Domain Independent Temporal Reasoning with Recurring Events", *Computational Intelligence* (to appear), 1997.

[14] M. Niezette, J.-M. Stevenne: "An Efficient Symbolic Representation of Periodic Time", *Proc. First International Conf. on Information and Knowledge Management*, 1992.

[15] R. Snodgrass: "The TSQL2 Query Language". Kluver Academic Publ., 1995.

[16] M. Soo, R. Snodgrass: "Multiple Calendar Support for Conventianal Database Management Systems", *Proc. International Workshop on an Infrastructure for Temporal Databases*, June 1993.

[17] P. Terenziani: "Reasoning about Periodic Events", *Proc. TIME-95*, pp. 137-144, Melbourne, FL, April 1995.

[18] P. Terenziani: "Towards an Ontology dealing with Periodic Events", *Proc. ECAI'96*, 43-47, 1996.

[19] P. Terenziani: "Representing and Reasoning with Temporal Constraints about Periodic Events", *Tech.Rep. TR13-96*, Dipartimento di Informatica, Universita' di Torino, 1996.

[20] P. Terenziani: "Integrating calendar-dates and qualitative temporal constraints in the treatment of periodic events", accepted for publication in *IEEE Transactions on Knowledge and Data Engeneering*, 1997.

[21] L. Vila: "A Survey of Temporal Reasoning in Artificial Intelligence", *AI Comm. 7(1)*, 4-128, 1994.

[22] R. Wetprasit, A. Sattar, and L. Khatib, "Reasoning with sequences of Point Events, *Proc. TIME-96;* Chittaro, Goodwin, Hamilton and Montanari eds., IEEE Press, pp. 36-38, Los Alamitos, CA, USA, 1996.