Reasoning about event granularity

Khalil BENKOULA (*), Salima BENBERNOU (*), Mahieddine DJOUDI (**)

(*) Université des Sciences et de la Technologie d'Oran
Institut d'Informatique, B. P. 1505 El M'nouar Oran, Algérie.

(**) Laboratoire SIC-IRCOM CNRS Département
d'Informatique de l'Université de Poitiers,
France.

ABSTRACT

In order to represent the event granularity a class of intervals, called c-interval, is proposed. It is defined in terms of a macro relations set that also allows to express the incomplete temporal assertions. Checking in first the consistency of such temporal assertions is an essential part of reasoning. An efficient algorithm is supplied to perform it. A second important task of reasoning is determining the additional temporal assertions. It can be computed with the constraints propagation approach. Another approach is used which requires the introduction of the notion of level to guide the search of temporal relations between pairs of intervals.

KEYWORDS: Artificial intelligence, temporal reasoning, knowledge representation, time interval, graph, event, granularity.

1. INTRODUCTION

The temporal reasoning is an important aspect in many artificial intelligence applications. For example, in planning the inconsistency (i.e. incoherence) of a temporal assertions set means that the choosen action is not appropriate for a given goal [16]. In prediction, that kind of reasoning allows to build under some hypothesis, the possible histories corresponding to the possible evolution of the modeled world [11].

James Allen [1] introduced a fundamental model based on the interval concept where the relative position between pairs of intervals is expressed by the thirteen basic relations which are mutually exclusives (see table 1); their set is called A_{13}.

Relation	Relation inverse	Interprétation
I1 before (<) I2	after (>)	I1 I2
I1 meets (m) I2	met-by (m-1)	I1 I2
I1 overlaps (o) I2	overlapped-by (o-1)	I1 I2
I1 starts (s) I2	started-by (s-1)	I1 I2
I1 during (d) I2	includes (d-1)	I1 I2
I1 finishes (f) I2	finished-by (f-1)	I2 I1
I1 equals (=) I2	equals (=)	I1 I2

Table 1: The basic relations of interval.

When temporal assertions between pairs of intervals are incomplete, then they are allowed to be a disjunction of the basic relations. These relations have been called macro relations by Freska [6]. The main temporal reasoning tasks include determining the consistency of such sets and deducing additional relations from those that are given. A constraints propagation algorithm has been developed by Allen [1] for performing these tasks which are NP-complete problem [15]. In Vilain and Kautz's [15] framework the instant concept is used for representing such temporal assertions where the time points are linked by three basic possible relations: *precedes* (<), *same* (=) and *follows* (>). However, a set of interval relations can be translated into conjunctions of points relations between the endpoints of intervals. Algorithms have been developed by Van Beek [13] for determining the consistency in $O(n^2)$ time and computing the additional relations in $O(n^4)$ time where n is the number of points.

The framework based on intervals is general and formal. However, models that are more adapted to applications are required. In particular, it is often useful and necessary to represent events in different levels of details or granularities. Montanari [8] introduced the notion of macro event for formalizing the decomposition operation of an event into other events. The granularity has been also used by Nokel [10] for automatic recognizing of events. In this paper, we propose in first an interval class, called c-interval, in terms of sub-intervals and a set of macro relations. This set is defined in order to represent the event granularity and the incomplete temporal relations. After representing the temporal assertions in a c-interval graph, an efficient algorithm is supplied for determining the consistency of those assertions. Then we discuss the problem of computing the additional temporal assertions. To perform it requires the use of the constraints propagation approach. At the end the notion of level is introduced and as result the intervals are partitioned into totally ordered sets which guide the search of temporal relations.

2. A MODEL OF COMPOSED INTERVALS

Considering the same reality in different levels of details is a way that is often used for structuring a domain. A such hierarchisation improve often the performance of the manipulation of entities when they are focused on adequate levels [5]. Two kind of granularities are distinguished: *event* and *time granularity*. The event granularity depends on the given descriptions. On the other hand, the time granularity depends on the domain refers to. The focus of the present paper is the event granularity which considers both the distinctions that a framework can make and the distinctions that it can leave unspecified. Consider for instance the description "Benali take his lunch" which is decomposed into the following events:

eat,
speak and
drink coffee

where each sub-event can also be refined. Since an event is supposed to be an interval in the real axis, we can decompose it into sub-intervals and then represent it under different granularities. Each sub-interval will correspond to a component event. Note, that no temporal assertions between the sub-event are provided. So, a set of macro relations is introduced. A macro relation is a subset of the basic thirteen relations and is considered as the disjunction of its constituent relations, expressing then an incomplete temporal relation between pairs of intervals. The main macro relation is the *during* relation, abbreviated by the symbol "$\subset$", between an interval and its sub-intervals. It characterizes the decomposition function of an event in other events. In fact, a sub-event occurs during its composed event. Furthermore, it allows a description of events with some precisions and after the refinement of this description, more details are given. In the previous example, we represent this as: "Eat $\subset$ Lunch", "Speak $\subset$ Lunch", and "Coffee $\subset$ Lunch". A such relation often disposes time intervals in a hierarchical structure [1].

The macro relations introduced as follows:

$\subset = \{ s, f, d, = \}$ $\quad\quad \subset^{-1} = \{ s^{-1}, f^{-1}, d^{-1}, = \}$
$\alpha = \{ <, m \}$ $\quad\quad \alpha^{-1} = \{ >, m^{-1} \}$
$\beta = \{ <, >, m, m^{-1}, o, o^{-1}, s, s^{-1}, f, f^{-1}, d, d^{-1}, = \}$

allow to define a set noted by $A_8 = \{<, >, \alpha, \alpha^{-1}, \subset, \subset^{-1}, \beta, = \}$ which its members are called atomic relations. The α (resp. α^{-1}) relation means that an interval is before or meets (resp. after or met-by) another interval. If no temporal relation between two intervals is given, that means the whole set of relations β is possible. Then, we represent the unknown relations in the example as: "Speak β Coffee", "Eat β Coffee" and "Speak β Eat".

The time intervals used will be noted by $I_1, I_2, ..., I_n$ with I will be the set of intervals and I_c will be the set of c-intervals. After the notion of the macro relation introduced, we can define now formally the c-interval class.

Definition 1: c-interval
An c-interval I is an interval which is decomposed into other sub intervals I_i with:

$\forall i\ 1 \le i \le n\ \ I_i \subset I$ and

$\forall i, j\ 1 \le i \le n$ and $1 \le j \le n$ such that $i \ne j$ and one of the following relations is checked:

$I_i < I_j$
$I_i\ \alpha\ I_j$
$I_i\ \beta\ I_j$
$I_i = I_j$

In other words a c-interval is a time interval composed by other c-intervals or intervals which can be linked by one relation of the set A_8.
We note that the *before* (<), *before* or *meets* (α), *during* ($\subset$), *unknown* (β) and *equal* (=) relations are transitive.
To assert that one or two intervals are components of a time interval, two predicates are introduced as follows:

$\mathrm{COMP}(I_1, I) \Leftrightarrow I_1$ composes I
$\mathrm{MEMB}_I(I_1, I_2) \Leftrightarrow \mathrm{COMP}(I_1, I) \wedge \mathrm{COMP}(I_2, I)$

Formally an event decomposition may be viewed as a function that maps all c-intervals to the corresponding set of component intervals. For instance, the event "go to retreive a book" can be decomposed into the events "go to the lending library", "write an application" and "take the book".

3. TEMPORAL ASSERTIONS REPRESENTATION

Since the temporal relations are binary, we use a temporal graph $CG = < \mathsf{I} \cup \mathsf{I}_c, R >$, called c-interval graph, for representing those relations. It consists of a finite set of nodes ($\mathsf{I} \cup \mathsf{I}_c$) and a finite set of edges R labeled with atomic relations of the set noted by $A_5 = \{<, \alpha, \subset, \beta, =\}$ with:

$\forall I_1 \in \mathsf{I}_c, \forall I_2 \in (\mathsf{I} \cup \mathsf{I}_c)\ \mathrm{COMP}(I_2, I_1)$
$\quad\Leftrightarrow (I_2, \subset, I_1) \in R$
$\forall I_1, I_2 \in (\mathsf{I} \cup \mathsf{I}_c), \exists r \in A_5\ \ (I_1, r, I_2) \in R$
$\quad\Rightarrow \exists I\ \ \mathrm{MEMB}_I(I_1, I_2)$

A path of a length n is a sequence of n triples $(I_0, r_1, I_1), (I_1, r_2, I_2), ..., (I_{n-1}, r_n, I_n)$ where I_i ($0 \le i \le n$) are nodes (i.e. c-intervals or intervals) and r_j ($1 \le j \le n$) are labels (i.e. temporal relations) on edges. In particular, the path is a cycle when I_0 and I_n are equal. If there exists an edge between all pairs of nodes then the graph is complete. As an example of representing temporal assertions in a c-interval graph, consider the lunch event shown in figure 2 where the description "Benali was speaking while eating his lunch, after he drank his coffee" is given.

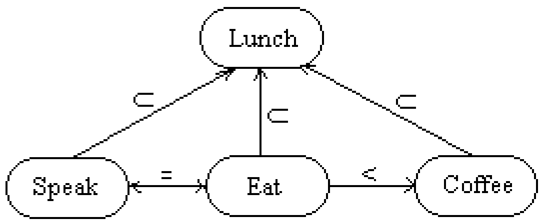

Figure 2: A representation of a composed event.

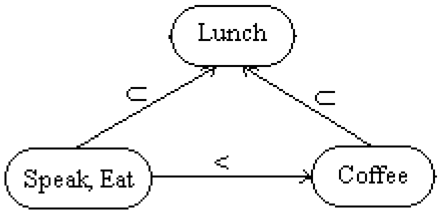

Figure 3: A reduced graph.

Since the *equal* relation is symetric, we represented it with a bi-directional edge. Furthermore, edges labeled with the *unknown* relation which gives no temporal informations have been suppressed. We can also suppress edges labeled with the *equal* relation by identifying all pairs of intervals that are necessarily equal and putting them into one node. For instance, putting the intervals "Speak" and "Eat" into one node would give the reduced graph shown in figure 3. So, a c-interval graph is an incomplete graph because the transitivity property of temporal relations is used in order to reduce the number of edges and nodes.

4. TEMPORAL ASSERTIONS CONSISTENCY

As in all knowledge representation systems, it is necessary to check the consistency of temporal assertions. To perform it requires the use of the c-interval graph. Consider the lunch event with adding now the description "Benali was eating while drinking his coffee, after he spoke". Representing those temporal assertions is shown in figure 4.

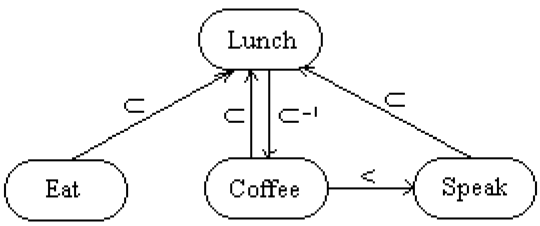

Figure 4: An example of an inconsistant graph.

By analysing only sentences, we can state that they are inconsitent: the lunch event is decomposed into the coffee sub-event and the opposite is false. In order to check the consistency of such temporal assertions represented in a c-interval graph, two kinds of paths are defined as follows:

Definition 2: $\subset$-path
In a c-interval graph we call a path, where all edges are labeled with the *during* ($\subset$) relation, an $\subset$-path.

Definition 3: α-path
In a c-interval graph we call a path, where at least one edge is labeled with the precedence (α or <) relation, an α-path.

If there exists a $\subset$-path from I_0 to I_n then we obtain $I_0 \subset I_n$ by applying the transitivity property of the *during* relation and what's that more the $\subset$-path is a cycle, then we have $I_n \subset I_0$. However, the *during* relation is symetric when it is restricted to the *equal* relation. Since the precedence (α or <) relations are irreflexive, the nonexistence of α-paths which are cycles involves the consistency of the c-interval graph.

Property 1
The c-interval graph is consistent if and only if
i) there exist a $\subset$-path of length n which is a cycle then $I_0 = I_n$,
ii) there not exist a α-path which is a cycle.

The previous description is inconsistent because the corresponding c-interval graph contains a cycle where one edge is labeled with the *before* (<) relation.
Thus, the task for determining the consistency of temporal assertions requires first identifying the strongly connected components (SCC) of the corresponding c-interval graph. In this step, an algorithm in [3] is used which computes them in O(max(n, m)) time where n is the number of nodes and m is the number of edges. After that, the label on each edge in the same SCC is compared with the relation < or α . So, the algorithm which performs these two previous steps is presented as follows:

INPUT: a c-interval graph CG = $< I \cup I_c, R >$
OUPUT: the answer about the consistency of CG

Check_Consistency(CG)
begin
1. Identifying the strongly connected components of CG
2. For each edge (I_i, r, I_j) $\in R$ do
   if *component(i)* = *component(j)* and (r = ' < ' or r = 'α')
   then
      return (" Inconsistent Graph ")
   endif
3. return (" Consistent Graph ")
end.

Identifying the SCCs maps each node I_i to a number noted by *component(i)* corresponding to its SCC. We have m edges that are partitioned in the different SCCs. A test is evaluated for each label on an edge. For this, a time in O(m) is required. Then, the global time needed for checking the consistency of a c-interval graph is in O(max(n, m)).

Property 2
Determining the consistency of a c-interval graph requires a time in O(max(n, m)), where n is the number of nodes and m is the number of edges.

Another approach which requires more time for determining the consitency of temporal assertions consists of the use of Allen [1] or Van beek's [13] algorithm. This gain of time is a consequence of the used relations. The last alternative is possible because all the atomic relations can be translated into conjunction of relations between pairs of points.

5. DEDUCING TEMPORAL ASSERTIONS

Given some temporal relations between pairs of intervals, we would like to have the ability to deduce the additional relations which are implicite. For instance, if an interval I_1 is during I_2, and I_2 is before I_3, then I_1 must be before I_3. This last relation added to the c-interval graph allows to deduce other relations. For performing automatically this reasoning task, two operations intersection and composition are defined.

5.1. Intersection

The intersection operation noted by '∩' supplies the restricted relation of two atomic relations which hold between the same pair of intervals. As an example, the *before* (<) relation is checked if the two relations < and α are explicitely given between a pair of intervals. Thus, an extract from the table of the intersection operation is shown in figure 5.

∩	<	α	=	β
α	<	α	∅	α
⊂	∅	∅	=	⊂
β	<	α	=	β

Figure 5: An extract of the intersection operation table.

Algorithmically, the intersection of two macro relations is computed by finding their common constituent relations.

5.2. Composition

The composition operation noted by '.' is defined between pairs of relations which linke three intervals I_1, I_2 and I_3. It is used for determining the relation between I_1 and I_2 where the relation between each one and the inteval I_3 is given. Thus, an extract from the table of the composition operation which is determined by using Allen's method [1] is shown in figure 6.

.	<	α	=	β
α	<	<	α	β
⊂	<	α	⊂	β
=	<	α	=	β

Figure 6: An extract of the composition operation table.

5.3. Deduction

The deduction of implicite relations can be viewed as the determination of the transitive closure of relations beween pairs of temporal intervals, with using the intersection operation and rules of the composition table. We consider for each pair of intervals (I_1, I_2) the restrictions made over relations between $I_1$and I_2 with other intervals until any relation can be restricted.

Given an event description: "In the night, it has run but after the ground became dry". In this sentence three expressions which can be associate to an interval are extracted: "In the night", "it has run" and "the ground became dry". Representing these temporal assertions in a sub c-interval graph is shown in figure 7.

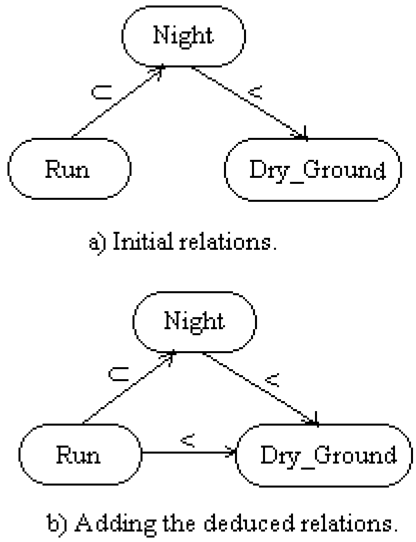

Figure 7: Representing the deduced relations

That scheme shows the importance of the deduction, for adding temporal relations in the graph even if they were not expressed explicitly in the description. If an interval is during a c-interval, then we can do or not the connection between it and another interval which is before or after its corresponding c-interval. This property which has been described by Dorn [3] can be checked with the table of the composition operation.

6. QUERY FOR TEMPORAL RELATIONS

Let us consider a consistent set of temporal assertions which are represented in a c-interval graph. We are interested in the problem of determining the strongest relation between two special intervals. The approach discussed above consists of computing the closure of the given temporal assertions. However, another efficient approach in space is based on graphs partitioned into a set of chains [4, 7], where a chain is a totally ordered set. Its main characteristic is their use of chains to guide the search of the strongest relation between pairs of intervals without determining all the additional relations.

We propose to partitione the set of intervals into chains where all intervals are compared by the *during* relation. A such chain is represented by a ⊂-path. To do that, levels of nodes are introduced. The level of a node is defined as the length of the longest ⊂-path from the node which is not connected to any other node by an edge labeled with the *during* relation. Consider the description "he ate, after he drank his coffee" added to the lunch event which is represented in figure 8.

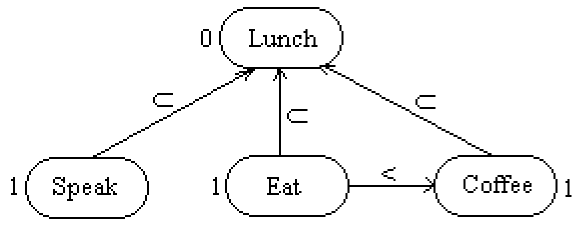

Figure 8: Levels of nodes in a c-interval graph.

Since the lunch event is not during other events, its corresponding level gets the value 0. On the other hand, the level of spoke , eat and coffee events that are linked to the lunch event with the *during* relation gets the value 1. An algorithm for computing levels of nodes in an acyclic c-interval graph is based on Gervini and Schubert's algorithm [7]. They have partitioned points into a set of chains which are sets of points totally ordered by the *precedes* or *same* (≤) relation Thus, the strongest relation between two intervals in the same chain can be obtained in a linear time. This is performed by only comparing their corresponding level. If two intervals are during the same c-interval and have the same level, then the strongest relation is derived by applying the transitivity property of relations that are labels of the α-path connecting them. When the α-path is nonexistent then an *unknown* relation is the strongest relation. The remaining case concerns two intervals which are not during the same c-interval. In fact, as a consequence of the property mentioned in section 5.3, the strongest relation is equal to the relation derived by replacing the interval that has the minimal level with its corresponding c-interval until conditions of the previous case hold.

7. CONCLUSION

We have presented a class of intervals for representing the event granularity. It is defined in terms of macro relations and sub-intervals. The advantage of representing temporal assertions in a c-interval graph have been noticed. So, checking the consistency of those assertions requires a linear time by using the algorithm supplied. Furthermore, the notion of level introduced allows to compute the strongest relation between pairs of intervals without determining all the additional relations.

REFERENCES

[1] James F. Allen (1983): "Maintaining Knowledge about Temporal Intervals". Communications of the ACM 1983, Vol 26, n 11, pages 832-843.

[2] James F.Allen (1984): "Towards a General Theory of Action and Time". Artificial Intelligence, pages 123-153.

[3] Sara Baase (1989): "Computer Algorithms, Introductions to Design and Analysis". Second Edition, addition-Wesley Publishing Company.

[4] Jürgen Dorn (1992): "Temporal Reasoning in Sequence Graphs". 3AI 1992, pages 735-740.

[5] Jérome Euzenat (1992): "Representation granulaire du temps". Intelligence artificielle, volume 7, n°3, page 329-361.

[6] Christian Freska (1992): "Temporal Reasoning based on semi-intervals". Artificial Intelligence, 54, pages 199-227.

[7] Alfonso Gervini et Lenhart Schubert (1993): "Efficient Temporal Reasoning through" Timegraphs. IJCAI 1993, pages 648-654.

[8] Angelo Montari, Enrico Main, Emanuele Ciapessoni, Elena Ratto (1992): "Dealing with Time granularity in the Event Calculus". Proceeding of the international conference on the fifth generation computer systems 1992, pages 702-712.

[9] Rober A. Morris et William D. Shaff et Linda Khatib (1993): "Path Consistency in a Network of Non-Convex Intervals". IJCAI 1993, pages 655-660.

[10] Klaus Nokel (1989): "Temporal Matching: Recognizing Dynamic Situations from Discrete Meeasurements". In proceeding of the 11th IJCAI, Detroit, 1255-1260.

[11] Wojciech Penczek (1988): "A Temporal Logic for Event Structures". Fundamenta Informaticae XI 1988, pages 297-326.

[12] Yoav Shoham et Drew McDermott (1988): "Problems in Formal Temporal Reasoning". Artificial Intelligence, 36, pages 49-61.

[13] Peter Van Beek (1990): "Exact and Approximate Reasoning about Qualitative Temporal Relations". Technical Report TR 90-29. August 1990, University of Alberta, Edmonton, Alberta, Canada.

[14] Peter Van Beek (1992): "Reasoning about Qualitative Temporal Information". Artificial Intelligence, 58, pages 297-326.

[15] Marc Vilain et Henry Kautz (1986): "Constraint Propagation Algorithm for Temporal Reasoning". Proceeding AAAI-1986, philadelphia, pages 377-382.

[16] R. Wylie et M. Kamel (1992): "Model Based Knowledge Organisation". Proceeding of the seventh international conference on application of artificial intelligence in engineering 1992, page 161-178.

[17] R. Wylie et M. Kamel (1992): "Model Based Knowledge Organisation : A Framework for Constructing High-level Control System". Expert systems with applications 1992, pages 285-296.