Temporal Query Processing With Indefinite Information

Peter van Beek
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
vanbeek@cs.ualberta.ca

Appears in: *Artificial Intelligence in Medicine,* 3:325-339, 1991.

Abstract

Time is an important aspect of information in medical domains. In this paper, we adopt Allen's influential interval algebra framework for representing temporal information. The interval algebra allows the representation of indefinite and incomplete information which is necessary in many applications. However, answering interesting queries in this framework has been shown to be almost assuredly intractable. We show that when the representation language is sufficiently restricted we can develop efficient algorithms for answering interesting classes of queries including: (i) determining whether a formula involving temporal relations between events is possibly true and necessarily true; and (ii) answering aggregation questions where the set of all events that satisfy a formula are retrieved. We also show, by examining applications of the interval algebra discussed in the literature, that our restriction on the representation language often is not overly restrictive in practice.

1 Introduction

Medical information and diagnostic systems must be able to represent and answer queries about temporal information about events such as a patient's medical history or the course of a disease. In many such medical applications, the ability to represent hierarchical, indefinite, and incomplete temporal information is necessary. Kahn et al. [9] show this to be true in expert diagnostic systems and in representing a patient's medical history. Hamlet and Hunter [7] also make this point using an account of a patient's symptoms. Kahn et al. [9] are led to develop an ad hoc approach to representing indefinite temporal information and they describe how, given the representation, some interesting classes of queries can be answered.

Another approach to representing temporal information is that used in temporal databases where events are stamped with start and end times and the granularity of the time stamp must be specified when the database is constructed [17, 16]. Temporal databases are useful in many applications and allow a broad range of query types. However, temporal databases lack the ability to represent indefinite and incomplete temporal information. Further, difficulties can arise if our temporal information is hierarchical or not all at the same level of detail, regardless of the granularity of the time stamp chosen. If a smaller granularity is chosen, we may have to represent more than we know about the times of an event when the times of that event are not known to the required level of precision. If a larger granularity is chosen, we may not be able to represent all that we know about the temporal relations between events. For example, if the granularity of the time stamp is the month an event occurred in, then we cannot specify that, for two events that occurred in the same month, one event preceded the other. The result is that we have lost information.

In this paper, we adopt Allen's [2] interval algebra framework for representing hierarchical and possibly indefinite and incomplete temporal information about the relations between events or intervals of time. The framework is influential and has been applied in such diverse areas as natural language processing [3], diagnosis [15], and medical expert systems [7, 8]. However, it has been shown that for Allen's representation, answering many of the queries we would like to be able to answer is NP-complete and thus almost assuredly intractable [25, 26].

We show that, if we sufficiently restrict the representation language, we can develop efficient algorithms for answering interesting classes of queries. We begin by reviewing previously known results for answering two fundamental queries: (i) find the feasible relations between all pairs of events, and (ii) determine whether the temporal information is consistent. We show, in turn, how these results can be used to develop new algorithms for (i) answering whether a formula involving temporal relations between events is possibly true and necessarily true; and (ii) answering aggregation questions where the set of all events that satisfy a formula are retrieved.

We also show, by examining the natural language, diagnostic, and artificial intelligence in medicine literature [18, 4, 15, 8, 7, 9], that our restriction on the representation language often is not overly restrictive in practice.

2 Representing Temporal Information

In this section we review Allen's framework for representing relations between events or intervals of time (we use events and intervals interchangeably). We then formalize the representation using networks of binary relations [14].

2.1 Allen's framework

There are thirteen **basic** relations that can hold between two intervals (see Fig. 1). In order to represent indefinite information, the relation between two intervals is allowed to be a disjunction of the basic relations. Sets are used to list the disjunctions. For example, the relation {m,o,s} between events A and B represents the disjunction,

$$(\text{A meets B}) \vee (\text{A overlaps B}) \vee (\text{A starts B}).$$

Let I be the set of all basic relations, {b,bi,m,mi,o,oi,s,si,d,di,f,fi,eq}. Allen allows the relation between two events to be any subset of I.

We use a graphical notation where vertices represent events and directed edges are labeled with sets of basic relations. As a graphical convention, we never show the edges (i,i), and if we show the edge (i,j), we do not show the edge (j,i). Any edge for which we have no explicit knowledge of the relation is labeled with I; by convention such edges are also not shown. We call networks with labels that are arbitrary subsets of I, interval algebra or **IA** networks.

As an example of representing temporal information using **IA** networks, consider the description of events shown in Fig. 2a. Not all of the temporal relations between events are explicitly or unambiguously given in the description. The first sentence tells us only that the interval of time over which Fred read the paper intersects with the interval of time over which Fred ate breakfast. We represent this as "paper {o,oi,s,si,d,di,f,fi,eq} breakfast." The second sentence fixes the relationship between some of the end points of the intervals over which Fred read his paper and over which Fred drank his coffee but it remains indefinite about others. We represent this as "paper {o,s,d} coffee."[1] But we also know that drinking coffee is a part of breakfast and so occurs during breakfast. We represent this as "coffee {d} breakfast." Finally, the information in the third sentence is represented as "walk {bi} breakfast." The resulting network is shown in Fig. 2a, where we have drawn a directed edge from "breakfast" to

[1] Another possibility is the relation {b,m,o,s,d}, since the scenario where reading the paper occurred entirely before drinking the coffee is not explicitly ruled out by the sentence.

Relation	Symbol	Inverse	Meaning
x before y	b	bi	
x meets y	m	mi	
x overlaps y	o	oi	
x starts y	s	si	
x during y	d	di	
x finishes y	f	fi	
x equal y	eq	eq	

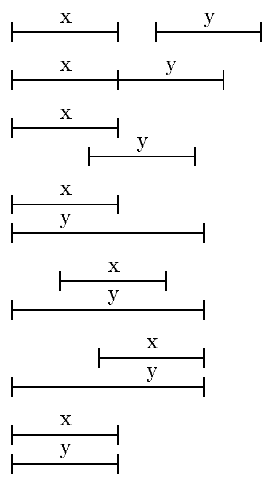


Figure 1: Basic relations between intervals

"walk" and so have labeled the edge with the inverse of the "bi" (after) relation. All edges not shown are labeled with I.

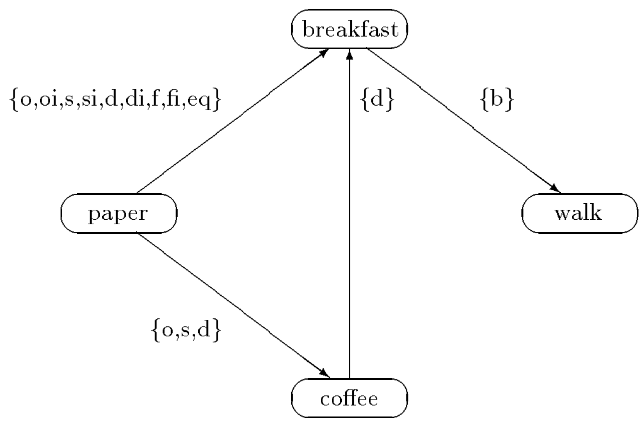
Vilain and Kautz [25, 26] show that answering the most fundamental questions of **IA** networks is NP-Complete and thus it is unlikely that efficient algorithms can be found for answering interesting queries. In this paper, we restrict the representation language and we show that, as a result, efficient algorithms can be devised. In particular, a restricted class of **IA** networks can be translated into conjunctions of inequalities, equalities, and disequalities between the endpoints of the intervals [25, 13]. For example, the interval relation {m,o,s} between events A and B can be represented as,

$$(\mathrm{A}^- < \mathrm{A}^+) \wedge (\mathrm{B}^- < \mathrm{B}^+) \wedge (\mathrm{A}^- \leq \mathrm{B}^-) \wedge (\mathrm{A}^+ \geq \mathrm{B}^-) \wedge (\mathrm{A}^+ < \mathrm{B}^+),$$

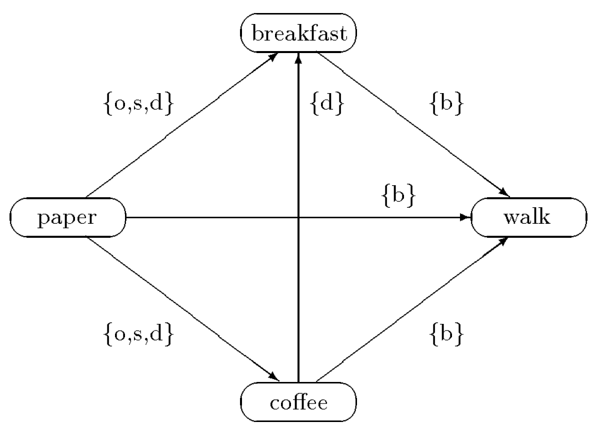
where A^- and A^+ represent the start and end points of interval A, respectively. Figure 3 shows a graphical representation of the translation.

In this paper, we further restrict ourselves to translations involving only the point-based relations $<$, $\leq$, $=$, $>$, $\geq$, and ?, where ? indicates we have no information about the relation between the points. That is, we do not allow the $\neq$ relation. We denote this restricted class of **IA** networks as **Simple IA** networks or **SIA** networks. Appendix A enumerates the relations, the subsets of I, that are allowed for **SIA** networks and also gives the translation into the point-based representation.

(a) Example: *Fred was reading the paper while eating his breakfast. He put the paper down and drank the last of his coffee. After breakfast he went for a walk.*



(b) Feasible relations:



(c) Consistent scenario:

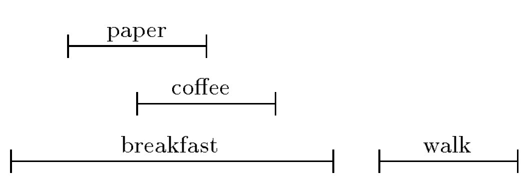


Figure 2: Representing qualitative relations between intervals

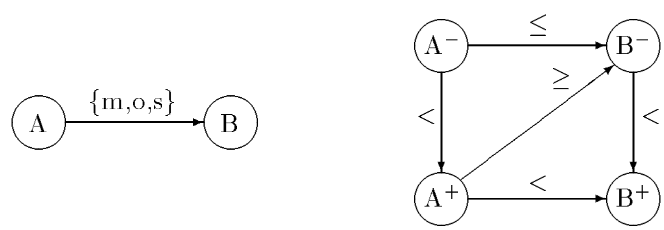


Figure 3: Example translation between interval- and point-based representation

The question arises as to whether the restricted representation language is still useful. Fortunately, the answer is yes as in many applications it is only this restricted language of relations that is ever used. For example, Almeida [4] and Song [18], in independent work on computer understanding of English narratives, both adopt Allen's framework but choose to use only relations that are in our restricted language **SIA**. Hamlet and Hunter ([7]; see also [8]) adopt Allen's framework for representing temporal information in medical expert systems but, with the exception of the disjointedness relation {b,bi,m,mi}, choose to use only relations in **SIA** (in their example, later temporal information is used to strengthen the disjointedness relation to {b,m} which is in **SIA**). Kahn et al. [9] devise a representation for temporal information for a medical diagnostic system. In Section 4.2, we show how this can be interpreted as an **SIA** network. Nökel [15] uses **SIA** networks in a diagnostic setting. With the exception of Nökel [15], it does not appear that the authors intentionally restricted their representation language or were aware of the computational advantages; rather, the relations used were simply the right ones for the task at hand.

2.2 Formalization of the representation

We formalize **IA** networks using networks of binary relations [14]. The advantage of this approach is that it allows us to use previously known algorithms and it allows us to say precisely what a query means and what constitutes a correct answer to a query.

A **network of binary relations** [14] is defined as a set X of n variables $\{X_1, X_2, ..., X_n\}$, a domain D_i of possible values for each variable, and binary relations between variables. A **binary relation**, C_{ij}, between variables X_i and X_j, is a subset of the Cartesian product of their domains that specifies the allowed pairs of values for X_i and X_j (i.e., $C_{ij} \subseteq D_i \times D_j$). For the networks of interest here, we require that $(x_j, x_i) \in C_{ji} \Leftrightarrow (x_i, x_j) \in C_{ij}$. An **instantiation** of the variables in X is an n-tuple $(x_1, x_2, \ldots, x_n)$, representing an assignment of $x_i \in D_i$ to X_i. A **consistent instantiation** of a network is an instantiation of the variables such that the relations between variables are satisfied. A network is **inconsistent** if no consistent instantiation exists.

An **IA network** is a network of binary relations where the variables represent time intervals, the domains of the variables are the set of ordered pairs of rational numbers $\{(s, e) \mid s < e\}$, with s and e representing the start and end points of the interval, respectively, and the binary relations between variables are represented implicitly by sets of the basic interval relations. For example, let $C_{ij} = \{\text{m,o}\}$ be the relation between variables X_i and X_j in some **IA** network. The set of allowed pairs of values for variables X_i and X_j is given by,

$$\{((s_i, e_i), (s_j, e_j)) \mid (s_i, e_i) \text{ meets } (s_j, e_j) \;\vee\; (s_i, e_i) \text{ overlaps } (s_j, e_j)\}.$$

The basic relations are disjoint. Hence, if an instantiation of variables X_i and X_j satisfies C_{ij}, then one and only one of the basic relations in C_{ij} is satisfied. A basic relation $B \in C_{ij}$ is **feasible** with respect to a network if and only if there exists a consistent instantiation of the network where B is satisfied. A network is **minimal** if every $B \in C_{ij}, i, j = 1, \ldots, n$ is feasible.

3 Fundamental Queries

In this section, we review results for two fundamental reasoning tasks: Given an **SIA** network, (i) find the feasible relations between all pairs of intervals, and (ii) determine whether the temporal information is consistent.

3.1 Feasible relations

Consider again the description of events shown in Fig. 2a. From the given temporal information we can make the simple inference that the "read paper" event must have occurred before the "go for walk" event. That is, the only feasible relation between those two events is that "read paper" occurred before "go for walk". All other temporal relations, such as the two events occurring simultaneously, are infeasible. Fig. 2b shows the feasible relations between all pairs of events. Many of the relations between events have been strengthened.

Kautz [25, 26] shows that finding the feasible relations of an **IA** network is NP-Complete. Allen [2] gives an $O(n^3)$ algorithm for finding an approximation to the feasible relations of an **IA** network. Van Beek and Cohen [20, 23] show that Allen's algorithm is exact for **SIA** networks. Van Beek and Cohen also show that a simpler and well-known generalization of the Floyd-Warshall algorithm [1] can be used to find the minimal network of an **SIA** network. (To use the Floyd-Warshall algorithm or Allen's algorithm, we need to specify the operations of intersection ($\cap$) and composition ($\cdot$); see [2, 12].)

FLOYD-WARSHALL(C)

for $k \leftarrow 1$ to n
  do for $i \leftarrow 1$ to n
    do for $j \leftarrow 1$ to n
      do $C_{ij} \leftarrow C_{ij} \cap (C_{ik} \cdot C_{kj})$

3.2 Consistent instantiation

Consider again the description of events shown in Fig. 2a. There are several possible scenarios that are consistent with the description of events. Fig. 2c shows one such scenario. Another possible consistent scenario is one where Fred starts to read his paper before he starts his breakfast. A consistent instantiation that would give the arrangement shown in Fig. 2c is, paper $\leftarrow (1, 3)$, breakfast $\leftarrow (0, 5)$, walk $\leftarrow (6, 7)$, and coffee $\leftarrow (2, 4)$, where we are equating the names of the vertices with the variables they represent.

As with finding the feasible relations, it is known that the task of finding a consistent instantiation of an **IA** network is NP-Complete [25, 26]. However, **SIA** networks can be translated into a set of linear inequalities of the form $X_i - X_j < 0$, $X_i - X_j \leq 0$, and $X_i - X_j = 0$ and thus finding a consistent instantiation reduces to finding a solution to a set of linear inequalities. Because of the special structure of the inequalities, a solution can be found using a shortest-path algorithm, such as the Floyd-Warshall algorithm (see [21] for more details). However, even more advantage can be taken of the special structure of the inequalities and van Beek [22] gives an $O(n^2)$ time algorithm for **SIA** networks (the algorithm also solves the more general problem of finding a consistent instantiation of a point-based network where the $\neq$ relation is allowed between points). Consistent-Instantiation is a version of that algorithm which decides whether a consistent instantiation exists.

CONSISTENT-INSTANTIATION(C)

1. Translate **SIA** network C into a point-based network P.
2. Identify the strongly connected components (SCCs) of P using only edges labeled with $\{<\}$, $\{<, =\}$, and $\{=\}$. Let $S_1, \ldots, S_m$ be the SCCs found.
3. **for** $i, j \leftarrow m$
4.   **do** $label \leftarrow \{<, =, >\}$
5.   **for** $v \in S_i, w \in S_j$
6.   **do** $label \leftarrow label \cap P_{vw}$
7.   **if** $label = \emptyset$
8.   **then return**(false)
9. **return**(true)

The intuition behind the algorithm and a detailed explanation and proof of correctness can be found elsewhere [22, 21]. Here we only briefly clarify Step 2. In Step 2, we partition the vertices into equivalence classes S_i, $1 \leq i \leq m$, such that vertices v and w are in the same equivalence class if and only if there is a path from v to w and a path from w to v using only edges labeled with $<$, $\leq$, or $=$ (in the algorithm the relations are represented using set notation, so, for example, $\leq$ is represented as $\{<, =\}$). Determining the equivalence classes is the same as identifying the strongly connected components of the graph and efficient algorithms are known (Tarjan [19]).

4 Complex Queries

In this section, we develop efficient query processing algorithms for (i) determining whether a formula involving temporal relations between events is possibly true and necessarily true; and (ii) answering aggregation questions where the set of all events that satisfy a formula are retrieved.

4.1 True-or-false questions

Possibly true. Let $E = \{e_1, \ldots, e_n\}$ be the set of all events and let ϕ be a variable-free formula involving relations between events built up from the logical connectives implication, equivalence, conjunction, disjunction, and negation. For example, let $\phi = (e_1\{\mathrm{d}\}e_2 \Rightarrow e_1\{\mathrm{o,s}\}e_3)$. We want to be able to answer true-or-false questions of the form,

> Is it possibly true that ϕ?

We write this as $\Diamond(\phi)$. The query is asked with respect to an **SIA** network C that represents the temporal information. Informally, we are asking whether there is some way of arranging the events that is consistent with our temporal information and also makes ϕ true. More formally, $\Diamond(\phi)$ is true if and only if there exists a consistent instantiation of C such that ϕ is also satisfied.

Procedure Possible gives an algorithm for answering $\Diamond(\phi)$ queries. The algorithm makes use of the procedure Consistent-Instantiation for determining whether an **SIA** network is consistent or inconsistent. The input to the algorithm is a variable-free formula, ϕ, and an **SIA** network represented as an $n \times n$ matrix C, where element C_{ij} is the label on edge (i, j).

To understand the algorithm, it is important to note that an **SIA** network, in logical notation, is the conjunction,

$$(e_1 R_{11} e_1 \wedge \cdots \wedge e_1 R_{1n} e_n) \wedge \cdots \wedge (e_n R_{n1} e_1 \wedge \cdots \wedge e_n R_{nn} e_n), \tag{1}$$

where R_{ij} is the relation between event e_i and e_j. The basic idea of the algorithm is now as follows. The query is first converted to an equivalent form called disjunctive normal form,

$$(e_i R_{ij} e_j \wedge \cdots \wedge e_k R_{kl} e_l) \vee \cdots \vee (e_l R_{lm} e_m \wedge \cdots \wedge e_m R_{mn} e_n). \tag{2}$$

Of course, for the query to be true, one of the disjuncts must be true. For each disjunct, d_i, of Eqn. 2 in turn, we form the conjunction of d_i and Eqn. 1 and test whether a consistent instantiation exists. If a consistent instantiation exists, the query is possibly true. If, for every disjunct, a consistent instantiation does not exist, the query is false.

We now describe the algorithm in somewhat more detail. In Step 1 of the algorithm, we eliminate all occurrences of the operators $\Rightarrow$, $\Leftarrow$, and $\Leftrightarrow$ by rewriting ϕ as an equivalent formula using only the $\neg$, $\wedge$, and $\vee$ operators.

In Step 2, we distribute negations through until each negation applies to a single literal which will be of the form, $e_i R_{ij} e_j$. Because the basic relations are mutually exclusive and exhaustive, we can eliminate all negations by using the equivalence,

$$\neg(e_i R_{ij} e_j) \Leftrightarrow e_i(I - R_{ij})e_j,$$

where $(I - R_{ij})$ is the set difference between the set of all basic relations, I, and the set of basic relations between event e_i and event e_j.

In Step 3, we ensure that each relation is in **SIA** as procedure Consistent-Instantiation is only correct for **SIA** networks. It is, of course, advantageous to decompose any relation not in **SIA** into as few relations as possible that are in **SIA**. For example, suppose $R_{ij} = \{\text{b,bi,m,mi}\}$, which is not in **SIA**. An equivalent formulation is the disjunction, $e_i\{\text{b}\}e_j \vee e_i\{\text{m}\}e_j \vee e_i\{\text{mi}\}e_j \vee e_i\{\text{bi}\}e_j$, where all of the relations are now in **SIA**. However, a better decomposition would be the equivalent disjunction, $e_i\{\text{b}, \text{m}\}e_j \vee e_i\{\text{bi}, \text{mi}\}e_j$. A good decomposition of an R_{ij} can be accomplished as follows. The elements of the language **SIA** are ordered according to decreasing cardinality. A pass through the elements is then made, checking whether each element of **SIA** is a subset of the R_{ij} we wish to decompose. If it is, we remove it from R_{ij}.

After Step 3, the formula consists of disjunctions and conjunctions of literals. In Step 4, the formula is put in disjunctive normal form. In Steps 5–11, for each disjunct, d_i, of the formula, we form the conjunction of d_i and the **SIA** network and test whether a consistent instantiation exists.

Possible(C, F)

1. Eliminate all implications and equivalences from F.
2. Distribute negations in F.
3. Intersect each R_{ij} with C_{ij} and decompose any $R_{ij} \notin$ **SIA**.
4. Convert F to disjunctive normal form.
5. **for** each disjunct d in F
6.   **do** $W \leftarrow C$
7.   **for** each conjunct $e_i R_{ij} e_j$ in d
8.   **do** $W_{ij} \leftarrow W_{ij} \cap R_{ij}$
9.   **if** Consistent-Instantiation(W)
10.   **then return**(true)
11. **return**(false)

We assume for all the examples of this section, that the queries are asked with respect to the temporal information given in Fig. 2. Further, we assume that C has been made minimal, say by applying the Floyd-Warshall algorithm; that is, C is the matrix representation of the minimal **SIA** network shown in Fig. 2b.

Example 1. Let the query be, Is it possibly true that Fred started to read his paper before he started his breakfast and before he started his coffee? That is, we are asking whether it is possible that $(\mathrm{p}^- < \mathrm{b}^-) \wedge (\mathrm{p}^- < \mathrm{c}^-)$, where p^- is the start point of the interval of time over which Fred read his paper. In the interval-based representation, we have

$$\phi = \mathrm{p}\{\mathrm{b}, \mathrm{m}, \mathrm{o}, \mathrm{di}, \mathrm{fi}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{b}, \mathrm{m}, \mathrm{o}, \mathrm{di}, \mathrm{fi}\}\mathrm{c}.$$

The query is then $\diamond(\phi)$. In procedure Possible, Steps 1 and 2 do not change the formula. In Step 3, each $e_i R_{ij} e_j$ in the formula is intersected with C_{ij} to give,

$$\mathrm{p}\{\mathrm{o}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{o}\}\mathrm{c},$$

and, in this example, all of the relations are in **SIA**. The formula is already in disjunctive normal form (Step 4). The **SIA** network that results form applying Steps 5–10 is as shown in Fig. 2b except that the label on the edge from paper to breakfast is {o} and the label on the edge from paper to coffee is {o}. The resulting **SIA** network is consistent. Hence, $\diamond(\phi)$ is true. For example, paper $\leftarrow (1, 4)$, coffee $\leftarrow (3, 5)$, breakfast $\leftarrow (2, 6)$, and walk $\leftarrow (7, 8)$ is one possible consistent instantiation of the network which also satisfies ϕ.

Theorem 1 *Let ϕ be a variable-free formula involving only interval relations between events. Procedure Possible correctly determines the truth value of $\diamond(\phi)$.*

It is important to note that, while the queries we can ask are restricted to being variable-free, they are not restricted to only contain relations in **SIA**, but can contain relations outside of **SIA**. For example, $\diamond(\mathrm{p}\{\mathrm{b}, \mathrm{bi}\}\mathrm{w})$ is a valid query, where the relation {b,bi} is in **IA** but not in **SIA**.

In the rest of the paper, procedure Possible is used as the basis of other algorithms for answering complex queries, so some discussion of its complexity is in order. In Step 4, the conversion of a formula F of size k into disjunctive normal form can, in the worst case, result in a formula with $O(2^k)$ disjuncts (for example, if F is in conjunctive normal form). Hence, in the worst case, the number of calls to procedure Consistent-Instantiation can be exponential in the length of F. Our claim then that algorithm is efficient rests on the assumption that the length of an input formula F is bounded and small. In related work, Ladkin [11] gives an exponential time procedure for determining whether an arbitrarily quantified formula is consistent.

Necessarily true. Again, let $E = \{e_1, \ldots, e_n\}$ be the set of all events and let ϕ be a variable-free formula involving relations between events built up from the logical connectives. We also want to be able to answer true-or-false questions of the form,

Is it necessarily true that ϕ?

We write this as $\Box(\phi)$. Informally, we are asking whether it must be the case that ϕ is true; i.e., that there is no way of arranging the events that is consistent with our temporal information and also makes ϕ false. Again, the question is asked with respect to some **SIA** network C that represents the temporal information. More formally, $\Box(\phi)$ is true if and only if, in every consistent instantiation of C, ϕ is satisfied.

Procedure Necessary gives an algorithm for answering $\Box(\phi)$ queries. The algorithm is extremely simple given that the following equivalence exists between the two different classes of queries, $\Box(\phi) \Leftrightarrow \neg\Diamond(\neg\phi)$. Again, the input to the algorithm is a variable-free formula, ϕ, and an **SIA** network represented as an $n \times n$ matrix C.

NECESSARY(C, F)
1. **return**($\neg$POSSIBLE($C, \neg F$))

Example 2. Let the query be, Is it necessarily true that either reading the paper starts breakfast or reading the paper overlaps or starts drinking coffee? More formally, let $\phi = (\mathrm{p}\{\mathrm{s}\}\mathrm{b} \vee \mathrm{p}\{\mathrm{o,s}\}\mathrm{c})$. The query is $\Box(\phi)$, which is equivalent to $\neg\Diamond(\neg\phi)$. In procedure Possible, the negations are pushed through until all negations apply only to literals (Step 2),

$$\begin{aligned} \neg\phi &= \neg(\mathrm{p}\{\mathrm{s}\}\mathrm{b} \vee \mathrm{p}\{\mathrm{o,s}\}\mathrm{c}) \\ &= \neg(\mathrm{p}\{\mathrm{s}\}\mathrm{b}) \wedge \neg(\mathrm{p}\{\mathrm{o,s}\}\mathrm{c}) \\ &= \mathrm{p}\{\mathrm{b,bi,m,mi,o,oi,si,d,di,f,fi,eq}\}\mathrm{b} \wedge \\ &\qquad \mathrm{p}\{\mathrm{b,bi,m,mi,oi,si,d,di,f,fi,eq}\}\mathrm{c}. \end{aligned}$$

Each $e_i R_{ij} e_j$ in the formula is intersected with C_{ij} (Step 3) to give,

$$\mathrm{p}\{\mathrm{o,d}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{d}\}\mathrm{c}.$$

The relation {o,d} is not in **SIA** and is represented as a disjunction of relations that are in **SIA**,

$$(\mathrm{p}\{\mathrm{o}\}\mathrm{b} \vee \mathrm{p}\{\mathrm{d}\}\mathrm{b}) \wedge \mathrm{p}\{\mathrm{d}\}\mathrm{c}.$$

The formula is converted to disjunctive normal form (Step 4),

$$(\mathrm{p}\{\mathrm{o}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{d}\}\mathrm{c}) \vee (\mathrm{p}\{\mathrm{d}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{d}\}\mathrm{c}).$$

The first disjunct is $(\mathrm{p}\{\mathrm{o}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{d}\}\mathrm{c})$ (Step 5) and no consistent instantiation is found. The second disjunct is $(\mathrm{p}\{\mathrm{d}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{d}\}\mathrm{c})$ and a consistent instantiation is found. For example, paper $\leftarrow (3,4)$, coffee $\leftarrow (2,5)$, breakfast $\leftarrow (1,6)$, and walk $\leftarrow (7,8)$ is one possible consistent instantiation of the network which also satisfies $\neg\phi$. Hence, $\Diamond(\neg\phi)$ is true. Hence, $\Box(\phi)$ is false as ϕ is not satisfied in every consistent instantiation.

Example 3. Let the query be, Is it necessarily true that, if reading the paper overlaps or starts breakfast, then reading the paper overlaps or starts drinking coffee? More formally, let $\phi = (\mathrm{p}\{\mathrm{o,s}\}\mathrm{b} \Rightarrow \mathrm{p}\{\mathrm{o,s}\}\mathrm{c})$. The query is $\Box(\phi)$, which is equivalent to $\neg\Diamond(\neg\phi)$. In procedure Possible, the implication is eliminated by substituting an equivalent formula using only disjunction and negation (Step 1) and the negations are pushed through until all negations apply only to literals (Step 2),

$$\begin{aligned}\neg\phi &= \neg(\mathrm{p}\{\mathrm{o,s}\}\mathrm{b} \Rightarrow \mathrm{p}\{\mathrm{o,s}\}\mathrm{c}) \\ &= \neg(\neg(\mathrm{p}\{\mathrm{o,s}\}\mathrm{b}) \vee \mathrm{p}\{\mathrm{o,s}\}\mathrm{c}) \\ &= \mathrm{p}\{\mathrm{o,s}\}\mathrm{b} \wedge \neg(\mathrm{p}\{\mathrm{o,s}\}\mathrm{c}) \\ &= \mathrm{p}\{\mathrm{o,s}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{b,bi,m,mi,oi,si,d,di,f,fi,eq}\}\mathrm{c}.\end{aligned}$$

Each $e_i R_{ij} e_j$ in the formula is intersected with C_{ij} (Step 3) to give,

$$\mathrm{p}\{\mathrm{o,s}\}\mathrm{b} \wedge \mathrm{p}\{\mathrm{d}\}\mathrm{c}.$$

In this example, all of the relations are in **SIA** and the formula is already in disjunctive normal form (Step 4). The **SIA** network that results from applying Steps 5–10 is as shown in Fig. 2b except that the label on the edge from paper to breakfast is {o,s} and the label on the edge from paper to coffee is {d}. The resulting **SIA** network is inconsistent. Hence, $\Diamond(\neg\phi)$ is false. Hence, $\Box(\phi)$ is true as ϕ is satisfied in every consistent instantiation.

Theorem 2 *Let ϕ be a variable-free formula involving only interval relations between events. Procedure Necessary correctly determines the truth value of $\Box(\phi)$.*

4.2 Aggregation questions

Let $E = \{e_1, \ldots, e_n\}$ be the set of all events and let ϕ be a quantifier-free formula with one free (unquantified) variable involving relations between events built up from the logical connectives. For example, let $\phi = (x\{\mathrm{d}\}e_2 \vee x\{\mathrm{o,s}\}e_3)$. An important class of queries we want to be able to answer are aggregation queries of the form,

Retrieve the set of events for which ϕ is possibly (necessarily) true?

More formally, the answer to such a query is the set S such that,

$$S = \{x \mid (x \in E) \wedge O(\phi)\},$$

where O is one of the operators $\Diamond$ or $\Box$.

Procedure Retrieve gives a straightforward algorithm for answering aggregation queries. The input to the algorithm is an **SIA** network represented as a matrix C, a set of events E, and a quantifier-free formula with one free variable.

The argument E to Retrieve does not have to be the set of all known events. It could possibly be the set of events returned from a previous call to Retrieve. As well, we assume that sets of events can be formed and named when we initially represent our temporal information as an **SIA** network. For example, when representing a patient's medical history, a natural set of events to name is the visits the patient has made to the hospital. The algorithm is easily generalized to accept as arguments multiple sets of events and a formula with multiple free variables and to return the set of all tuples of events that satisfy the formula (an example is given below). Finally, we note that in an implementation of the procedure, we would be careful to not redo Steps 1–4 of procedure Possible each time through the for loop.

```
RETRIEVE(C, E, F)
1. S ← ∅
2. for each e ∈ E
3.     do  apply substitution to F
4.         if (operator(F) = ◇ ∧ POSSIBLE(C, F)) ∨
              (operator(F) = □ ∧ NECESSARY(C, F))
5.             then S ← S ∪ e
6. return(S)
```

Examples of the algorithm are drawn from information about a patient's medical history. Kahn et al. [9] develop an ad hoc approach for representing hierarchical and indefinite temporal information for a medical diagnostic system (see Fig. 4). Two kinds of links are used in their temporal network. A difficulty with their approach is that it is not clear exactly what a link means and what the absence of a link means. The temporal information in the network of Fig. 4 can be represented as an **SIA** network with the advantages that we know exactly what a link means and we have a richer representation language. In the **SIA** network, the absence of an arc means the relation between the two events is labeled with I, the set of all basic relations. The dashed arrows in the original network are labeled with the before relation in the **SIA** network and the solid arrows are labeled with the during relation with two exceptions. The exceptions result from the fact that hierarchical information is being represented. The chemotherapy interval consists of three subintervals cycle1, cycle2, and cycle3. The exceptions are (i) the relation between the interval cycle1 and the interval chemotherapy should be the starts relations; and (ii) if cycle3 is the last chemotherapy treatment, the relation between cycle3 and chemotherapy should be the finishes relation; otherwise the relation is the during relation. Similar reasoning applies to the radiation therapy interval which consists of three subintervals representing three separate radiation treatments. (See [2, 10] for discussions about representing hierarchies of events in **IA** networks.)

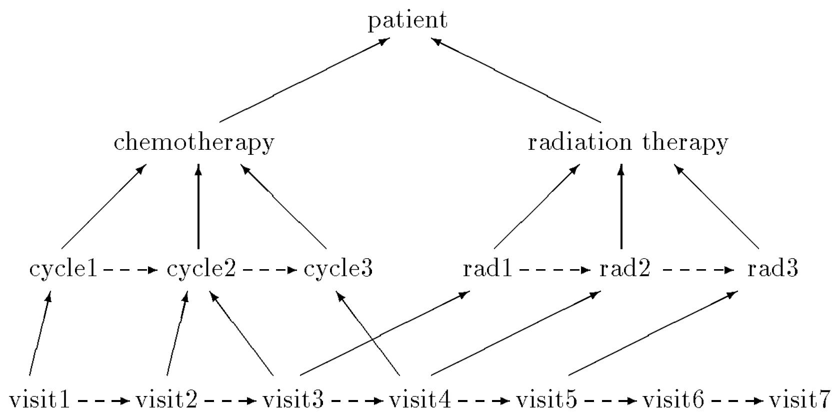


Figure 4: A patient's medical history (Kahn et al. [9])

To complete the temporal information used in our examples, we adopt a suggestion by Vilain [24] and use dates as time interval constants (see Fig. 5). This allows queries about the relationship between an event and a date as well as the relationship between two events.

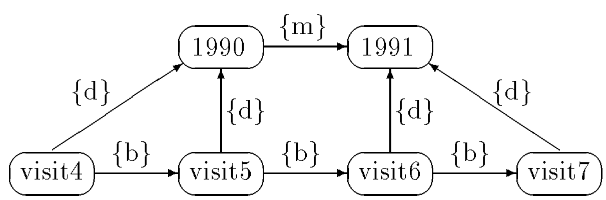


Figure 5: Representing information about dates

Example 4. Let the query be, "Retrieve all visits that necessarily occurred during 1991." The answer to the query is the set S, such that,

$$\begin{aligned} S &= \{x \mid (x \in \text{visits}) \wedge \Box(x\{\text{d}\}1991)\} \\ &= \{\text{visit6}, \text{visit7}\}. \end{aligned}$$

Example 5. Let the query be, "Retrieve all pairs of chemotherapy cycles and radiation treatments that necessarily intersect." We assume in this example that the algorithm has been generalized to accept multiple sets of events and a formula with multiple free variables and to return the set of all tuples of events that satisfy the formula. We also assume that cycles = {cycle1, cycle2, cycle3} and that rads = {rad1, rad2, rad3}. The answer to the query is the set S, such that,

$$
\begin{aligned}
S &= \{(x, y) \mid (x \in \text{cycles}) \wedge (y \in \text{rads}) \wedge \\
&\qquad\qquad \Box(x\{\text{o}, \text{oi}, \text{s}, \text{si}, \text{d}, \text{di}, \text{f}, \text{fi}, \text{eq}\}y)\} \\
&= \{(\text{cycle2}, \text{rad1}), (\text{cycle3}, \text{rad2})\}.
\end{aligned}
$$

In practice, each node in the **SIA** network would contain additional information about the associated event. For example, cycle1, a chemotherapy treatment event, could contain, among other things, dosage information. Additional routines would then be defined to, for example, print parts of that information or find the maximum of a numeric value, once the desired set of events had been retrieved.

Theorem 3 *Let a query be $O(\phi)$ where O is one of the operators $\Diamond$ or $\Box$, and ϕ is a quantifier-free formula with one free variable involving only interval relations between events. Procedure Retrieve correctly determines the set of events that satisfy $O(\phi)$.*

5 Conclusions

Representing and reasoning about hierarchical, indefinite, and incomplete temporal information is important in many medical information and diagnostic systems. In this paper, we adopted a restricted version of Allen's interval algebra framework for representing such information. We showed how we could answer two important classes of temporal queries: (i) true-or-false questions as to whether a formula involving temporal relations between events is possibly true and necessarily true; and (ii) aggregation questions where the answer to the query is the set of all events that satisfy a formula.

In current work, we are looking at algorithms for two additional classes of queries: (i) questions about whether a set of events can be linearly ordered and, if so, return the first, last, or kth event; and (ii) hypothetical or "what if" questions where, assuming that a formula involving temporal relations between events is true, the consequences of the formula are determined. Ordering questions should prove useful in contexts such as the patient's medical history shown in Fig. 4 where we want to be able to ask questions such as "What was the dosage given during the last visit" (see [9] for some related work on this problem). Hypothetical questions should prove useful in diagnostic reasoning (see [7] for some related work on this problem).

Allen's framework allows the representation of qualitative relations between events and between events and calendar dates, but does not allow the representation of quantitative information about the duration of events. Dechter et al. [6] and Dean [5] give frameworks for representing quantitative temporal information. For future work we intend to look at answering queries in these two alternative frameworks.

Acknowledgements. Financial assistance was received from the Central Research Fund of the University of Alberta and the Natural Sciences and Engineering Research Council of Canada.

A Appendix

In this appendix, we enumerate **SIA** and the translation into the point-based representation. A^- and A^+ represent the start and end points of interval A, and $A^- < A^+$ and $B^- < B^+$ are true for every translation. Inverses are not shown; for example, {bi} is also in **SIA**.

	A^-B^-	A^-B^+	A^+B^-	A^+B^+		A^-B^-	A^-B^+	A^+B^-	A^+B^+
{b}	?	?	$<$	?	{b,m,o,s}	$\leq$	?	?	$<$
{m}	?	?	$=$	?	{b,m,o,fi}	$<$	?	?	$\leq$
{o}	$<$	?	$>$	$<$	{m,o,s,d}	?	?	$\geq$	$<$
{s}	$=$	?	?	$<$	{m,o,di,fi}	$<$	?	$\geq$	?
{d}	$>$	?	?	$<$	{o,s,fi,eq}	$\leq$	?	$>$	$\leq$
{f}	$>$	?	?	$=$	{s,d,f,eq}	$\geq$	?	?	$\leq$
{eq}	$=$	?	?	$=$	{b,m,o,s,d}	?	?	?	$<$
{b,m}	?	?	$\leq$	?	{b,m,o,di,fi}	$<$	?	?	?
{m,o}	$<$	?	$\geq$	$<$	{m,o,s,fi,eq}	$\leq$	?	$\geq$	$\leq$
{o,s}	$\leq$	?	$>$	$<$	{b,m,o,s,fi,eq}	$\leq$	?	?	$\leq$
{o,fi}	$<$	?	$>$	$\leq$	{o,s,d,f,fi,eq}	?	?	$>$	$\leq$
{s,d}	$\geq$	?	?	$<$	{o,s,si,di,fi,eq}	$\leq$	?	$>$	?
{s,eq}	$=$	?	?	$\leq$	{m,o,s,d,f,fi,eq}	?	?	$\geq$	$\leq$
{d,f}	$>$	?	?	$\leq$	{m,o,s,si,di,fi,eq}	$\leq$	?	$\geq$	?
{f,eq}	$\geq$	?	?	$=$	$I-${b,m,o,s,d}	?	?	?	$\geq$
{b,m,o}	$<$	?	?	$<$	$I-${b,m,o,di,fi}	$\geq$	?	?	?
{m,o,s}	$\leq$	?	$\geq$	$<$	$I-${b,bi,m,mi}	?	$<$	$>$	?
{m,o,fi}	$<$	?	$\geq$	$\leq$	$I-${b,bi,m}	?	$\leq$	$>$	?
{o,s,d}	?	?	$>$	$<$	$I-${b,m}	?	?	$>$	?
{o,di,fi}	$<$	?	$>$	?	$I-${b,bi}	?	$\leq$	$\geq$	?
{s,si,eq}	$=$	?	?	?	$I-${b}	?	?	$\geq$	?
{f,fi,eq}	?	?	?	$=$	I	?	?	?	?

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Comm. ACM*, 26:832–843, 1983.

[3] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[4] M. J. Almeida. *Reasoning about the Temporal Structure of Narrative*. PhD thesis, State University of New York at Buffalo, 1987. Available as: Department of Computer Science Technical Report 87-10.

[5] T. Dean. Using temporal hierarchies to efficiently maintain large temporal databases. *J. ACM*, 36:687–718, 1989.

[6] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 83–93, Toronto, Ont., 1989. A longer version is to appear in *Artificial Intelligence* .

[7] I. Hamlet and J. Hunter. A representation of time for medical expert systems. In *Proceedings of the European Conference on Artificial Intelligence in Medicine*, pages 112–119, Marseilles, 1987. Available as: Lecture Notes in Medical Informatics 33. Springer-Verlag.

[8] J. Hunter, N. Gotts, I. Hamlet, and I. Kirby. Qualitative spatial and temporal reasoning in cardiac electrophysiology. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 157–166, London, 1989. Available as: Lecture Notes in Medical Informatics 38. Springer-Verlag.

[9] M. G. Kahn, J. C. Ferguson, E. H. Shortliffe, and L. M. Fagan. Representation and use of temporal information in oncocin - cancer therapy planning program. In M. K. Chytil and R. Engelbrecht, editors, *Expert Medical Systems*, pages 35–44. Sigma Press, 1987.

[10] J. A. Koomen. Localizing temporal constraint propagation. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 198–202, Toronto, Ont., 1989.

[11] P. B. Ladkin. Satisfying first-order constraints about time intervals. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 512–517, Saint Paul, Minn., 1988.

[12] P. B. Ladkin. Constraint reasoning with intervals: A tutorial, survey and bibliography. Technical Report TR-90-059, International Computer Science Institute, Berkeley, Calif., 1990.

[13] P. B. Ladkin and R. Maddux. On binary constraint networks. Technical report, Kestrel Institute, Palo Alto, Calif., 1988.

[14] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Inform. Sci.*, 7:95–132, 1974.

[15] K. Nökel. Temporal matching: Recognizing dynamic situations from discrete measurements. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1255–1260, Detroit, Mich., 1989.

[16] R. Snodgrass. The temporal query language TQuel. *ACM Transactions on Database Systems*, 12:247–298, 1987.

[17] R. Snodgrass and I. Ahn. Temporal databases. *IEEE Computer*, 19:35–42, 1986.

[18] F. Song and R. Cohen. The interpretation of temporal relations in narrative. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 745–750, Saint Paul, Minn., 1988.

[19] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1:146–160, 1972.

[20] P. van Beek. Approximation algorithms for temporal reasoning. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1291–1296, Detroit, Mich., 1989.

[21] P. van Beek. *Exact and Approximate Reasoning about Qualitative Temporal Relations*. PhD thesis, University of Waterloo, 1990. Available as: Department of Computing Science Technical Report TR-90-29, University of Alberta.

[22] P. van Beek. Reasoning about qualitative temporal information. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 728–734, Boston, Mass., 1990.

[23] P. van Beek and R. Cohen. Exact and approximate reasoning about temporal relations. *Computational Intelligence*, 6:132–144, 1990.

[24] M. Vilain. A system for reasoning about time. In *Proceedings of the Second National Conference on Artificial Intelligence*, pages 197–201, Pittsburgh, Pa., 1982.

[25] M. Vilain and H. Kautz. Constraint propagation algorithms for temporal reasoning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 377–382, Philadelphia, Pa., 1986.

[26] M. Vilain, H. Kautz, and P. van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. In D. S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan-Kaufman, 1989.