An Interval Algebra for Indeterminate Time

Wes Cowley
Department of Computer Science and Engineering
University of South Florida
P.O. Box 280138
Tampa, FL 33682-0138
wcowley@acm.org

Dimitris Plexousakis
Department of Computer Science
University of Crete & ICS-FORTH
P.O. Box 2208, Heraklion
Greece 71409
dp@csd.uch.gr

Abstract

Temporal indeterminacy is an inherent problem which arises when capturing and manipulating temporal data in many application areas. As such, representation and manipulation of timestamps with indeterminacy is a requirement for these applications. We present an extension of Allen's thirteen interval relationships to indeterminate temporal intervals based on a novel representation for indeterminate timestamps. The timestamps can be derived from and translated to interval constraints. We provide a set of simple and useful operators for manipulating both convex and non-convex indeterminate intervals represented by these timestamps.

Introduction

Temporal indeterminacy exists when one does not know precisely either the location of an event on the time line or its duration. The temporal knowledge about the event is indefinite or incomplete. This is an inherent problem when reasoning in many application domains. Planning systems are sometimes faced with the infeasibility of completely modeling all aspects of a situation. Knowledge representation and reasoning systems may be faced with insufficient information to completely determine a desired time interval. The knowledge about the time interval is indeterminate: one may know a period of time the interval definitely covers but there may also be a period of time into which the interval may extend.

The prevalent method for handling temporal indeterminacy in the temporal reasoning community is to use a temporal constraint network (TCN) on intervals (Allen 1983). The TCN allows the reasoner to arrive at a disjunction of possible relationships between pairs of intervals. For example, Anger, et al. describe a temporal knowledge base using this approach (Anger *et al.* 1988). The knowledge of the location and duration of an event on the time line is encoded in the relationships between that event and others in the network.

We examine a novel representation for indeterminate intervals based on the endpoints of the intervals' determinate and indeterminate regions and define a set of operators, which are both useful and non-complex, for manipulating indeterminate timestamps. Conversions between interval constraints in the form of the thirteen interval relationships (Allen 1983) and the indeterminate timestamps is straightforward. We use the indeterminate intervals to extend the traditional interval relationships to allow for indeterminacy. To the best of our knowledge, this is the first effort in this direction.

Our approach differs from TCNs in that the knowledge of an interval's position and duration is encoded in the interval's representation rather than in relationships between interval, allowing for non-complex manipulation of the knowledge. One application of this is in temporal integrity constraints for knowledge bases such as Telos (Plexousakis 1995).

Another method associates characteristic functions with intervals to represent the level of possibility that any given point is actually contained in the interval (Bouzid & Mouaddib 1998). In this approach, the characteristic function of intervals are modified as more knowledge is gained and schedules evolve. A similar method, in the temporal database field, describes a timestamp with a probability function (Dyreson & Snodgrass 1998). The probability and characteristic function approach is both more flexible and more complex than that described here as it allows for reasoning over the likelihood that a point is in the interval or not. In our method a point is either definitely in the interval or possibly in the interval.

In another approach, Gadia, et al propose a set oriented representation which supports both indeterminacy and incompleteness with a three-valued logic (Gadia, Nair, & Poon 1992). Their work is the closest to the indeterminate intervals described here, but provides a different set of operators. They do not describe a method for translating between interval constraint notation and their representation, complexity of the operators, or algorithms. Our work does not examine temporally missing values as Gadia does.

Koubarakis examines indeterminacy in the context of temporal constraint databases, where temporal indeterminacy is described by local and global constraints on variables representing the end points of the intervals (Koubarakis 1994; 1997). This is similar to the interval oriented TCN approach in that knowledge of a time interval is encoded in the constraints. The purpose in this case is less oriented toward deduction and more toward finite representation of potentially infinite domains.

In the remainder of this paper we will first describe our assumed temporal structure and chosen representation for indeterminate intervals followed by a discussion of an extension of the interval relationships to handle indeterminacy. Finally, we will give our conclusions and avenues for future work.

Valid Interval Stamps

Motivation Indeterminate time may arise in a number of ways and is nearly unavoidable when dealing with the valid time of facts in a knowledge base. We are interested in associating a valid time period with facts in order to support a temporal integrity constraint implementation, such as in, e.g., (Plexousakis 1995; Cowley 1999) in which constraints include combinations of the thirteen interval relationships (Allen 1983). As such, it is appropriate that the valid time periods are also specified using interval relationships. Because most of these relationships do not precisely constrain the time periods involved, the resulting time stamps are indeterminate. Thus, we must be able to derive indeterminate valid time stamps from interval constraints, manipulate those indeterminate intervals, and translate the results back to interval constraints.

Points We assume a linear discrete time line (Benthem 1991) bounded by the range of the underlying integer type in which temporal points are represented. We further assume distinct elements ∞, $-\infty$, and nil. With the exception of the incomparable element nil, which represents a non-specified value, points are totally ordered by $<$. The relationships $<$, $=$, and $\leq$ between two points carry the expected meaning. More formally, we assume a temporal structure $\mathcal{T}$ which is comprised of a set of points $\mathcal{P} = \{\ldots, p_{-1}, p_0, p_1, \ldots\}$ and a relation $<$. The set $\mathcal{P}' = \mathcal{P} \cup \{-\infty, \infty, \texttt{nil}\}$ is isomorphic to the set $\mathcal{Z}' = \mathcal{Z} \cup \{-\infty, \infty, \texttt{nil}\}$. We define a one-to-one and onto mapping function, $\iota : \mathcal{P}' \to \mathcal{Z}'$, and its inverse as follows:

$$\iota(p) = \begin{cases} p & \text{if } p = -\infty, \infty, \text{ or } \texttt{nil} \\ i & \text{for } p_i,\ i \in \mathcal{Z} \end{cases}$$

$$\iota^{-1}(i) = \begin{cases} i & \text{if } i = -\infty, \infty, \text{ or } \texttt{nil} \\ p_i & \text{for } i \in \mathcal{Z} \end{cases}$$

The relation $<$ between members of the set $\mathcal{P}$ is defined as $p_1 < p_2 \Leftrightarrow \iota(p_1) < \iota(p_2)$ and can be extended to the special elements $-\infty$ and ∞ by observing that $\forall p \in \mathcal{P}\ -\infty < p < \infty$. Finally, $<$ is undefined if either operand is nil. Similar translations give the meanings for $=$ and $\leq$. Note that from the definition of ι, $p_i = p_j \Rightarrow i = j$.

The following functions permit us to determine which of a pair of points occurs earlier or later on the time line.

Definition 1. We define The functions $\mathrm{min_p}$, $\mathrm{max_p}$, $\mathrm{min_{vp}}$, and $\mathrm{max_{vp}}$: $\mathcal{P}' \times \mathcal{P}' \to \mathcal{P}'$ as follows:

$$\mathrm{min_p}(p_1, p_2) = \begin{cases} \texttt{nil} & \text{if } p_1 = \texttt{nil} \vee p_2 = \texttt{nil} \\ p_1 & \text{if } p_1 < p_2 \\ p_2 & \text{otherwise} \end{cases}$$

$$\mathrm{max_p}(p_1, p_2) = \begin{cases} \texttt{nil} & \text{if } p_1 = \texttt{nil} \vee p_2 = \texttt{nil} \\ p_1 & \text{if } p_2 < p_1 \\ p_2 & \text{otherwise} \end{cases}$$

$$\mathrm{min_{vp}}(p_1, p_2) = \begin{cases} \texttt{nil} & \text{if } p_1 = \texttt{nil} \wedge p_2 = \texttt{nil} \\ p_1 & \text{if } p_2 = \texttt{nil} \vee p_1 < p_2 \\ p_2 & \text{otherwise} \end{cases}$$

$$\mathrm{max_{vp}}(p_1, p_2) = \begin{cases} \texttt{nil} & \text{if } p_1 = \texttt{nil} \wedge p_2 = \texttt{nil} \\ p_1 & \text{if } p_2 = \texttt{nil} \vee p_2 < p_1 \\ p_2 & \text{otherwise} \end{cases}$$

$\mathrm{min_p}$ ($\mathrm{max_p}$) chooses the earliest (latest) operand if both are not nil. Where exactly one operand is nil, $\mathrm{min_{vp}}$ and $\mathrm{max_{vp}}$ return the other point.

Given a point p, we need to refer to its next and previous points on the time line.

Definition 2. The functions previous and next, each with signature $\mathcal{P}' \to \mathcal{P}'$, are defined as follows:

$$\mathrm{previous}(p) = \begin{cases} p & \text{if } p \in \{-\infty, \infty, \texttt{nil}\} \\ \iota^{-1}(\iota(p) - 1) & \text{otherwise} \end{cases}$$

$$\mathrm{next}(p) = \begin{cases} p & \text{if } p \in \{-\infty, \infty, \texttt{nil}\} \\ \iota^{-1}(\iota(p) + 1) & \text{otherwise} \end{cases}$$

Determinate Intervals In order to discuss periods of time, as opposed to instants, we use intervals.

Definition 3. A *convex interval* is a set of consecutive points. An interval I can be represented by its lowest and highest points $\langle I_s, I_e \rangle$, $I_s \leq I_e$, $I_s = \texttt{nil} \Leftrightarrow I_e = \texttt{nil}$. We say that $p \in I$ iff $I_s \leq p \leq I_e$. The empty interval, which contains no point, is represented by $\langle \texttt{nil}, \texttt{nil} \rangle$ or $\emptyset$. We say that a convex interval is *infinite* if one or both endpoints is ∞ or $-\infty$. $\mathcal{I}$ refers to the set of all convex intervals. Points may be implicitly converted to intervals by the function *interval*: $\mathcal{P} \to \mathcal{I}$, defined as $\mathrm{interval}(p) = \langle p, p \rangle$.

We will use the 13 basic interval relationships (Allen 1983) for comparing intervals for ordering and inclusion. The concept of intersection of convex intervals is important. This carries the same meaning as in set theory, namely that there is at least one point in common.

Definition 4. Two convex intervals I_1 and I_2 *intersect* if $I_{1_s} \leq I_{2_e} \wedge I_{2_s} \leq I_{1_e}$.

Note that the empty interval, $\langle \texttt{nil}, \texttt{nil} \rangle$ does not intersect with any interval due to the incomparability of nil. The infinite interval, $\langle -\infty, \infty \rangle$ intersects with all non-empty intervals. We also need the concept of adjacency of convex intervals, which is defined next.

Definition 5. We say that two intervals I_1 and I_2 are *adjacent* if $\mathrm{next}(I_{1_e}) = I_{2_s} \vee \mathrm{next}(I_{2_e}) = I_{1_s}$.

We will also need the concept of non-convex intervals to talk about sets of non-consecutive points.

Definition 6. A *non-convex interval* I is a set of convex intervals I_i which neither intersect nor are adjacent.

Indeterminate Intervals A *valid interval stamp* (VIS) is an extension of the interval concept. A VIS describes the set of points which are definitely in the interval as well as those which *may* be in the interval. Through this mechanism we can represent the indeterminacy inherent in interval constraints.

Definition 7. A *convex valid interval stamp* (CVIS) is a 4-tuple $V = \langle I_s, D_s, D_e, I_e \rangle$ associated with a fact in a knowledge base. The points are restricted so that $D_s = \mathtt{nil} \Leftrightarrow D_e = \mathtt{nil}$ and $D_s = \mathtt{nil} \Rightarrow (I_s = \mathtt{nil} \Leftrightarrow I_e = \mathtt{nil})$. D_s, D_e, if not $\mathtt{nil}$, are time stamps marking the end points of the determinate interval D; that period of time during which the associated fact is true. I_s and I_e, if not $\mathtt{nil}$, represent the extended period of time before and after the determinate interval, respectively, during which the associated fact *may* be true. The indeterminate interval is the non-convex interval $I = \{I_L, I_H\}$ where I_L and I_H are:

$$I_L = \begin{cases} \emptyset & \text{If } I_s = \mathtt{nil} \\ \langle V.I_s, V.I_e \rangle & \text{If } D = \emptyset \\ \langle V.I_s, \mathrm{previous}(V.D_s) \rangle & \text{otherwise} \end{cases}$$

$$I_H = \begin{cases} \emptyset & \text{If } I_e = \mathtt{nil} \vee D = \emptyset \\ \langle \mathrm{next}(V.D_e), V.I_e \rangle & \text{otherwise} \end{cases}$$

A CVIS for which both I_s and I_e are $\mathtt{nil}$ is called *fully determinate*, while one with both D_s and D_e being $\mathtt{nil}$ is called *fully indeterminate*. The empty VIS is represented as $\langle \mathtt{nil}, \mathtt{nil}, \mathtt{nil}, \mathtt{nil} \rangle$ or $\emptyset$. We will use $\mathcal{CV}$ to refer to the set of all CVISs. An interval I can be converted implicitly to a CVIS with the function $\mathrm{cvis} : \mathcal{I} \rightarrow \mathcal{CV}$, defined as $\mathrm{cvis}(I) = \langle \mathtt{nil}, I_s, I_e, \mathtt{nil} \rangle$

Example 1. Figure 1a shows the CVIS $\langle 1, 2, 4, 6 \rangle$. This contains the points $\{2, 3, 4\}$ determinately and $\{1, 5, 6\}$ indeterminately. Figure 1b shows the same period covered by the fully indeterminate CVIS $\langle 1, \mathtt{nil}, \mathtt{nil}, 6 \rangle$. Finally, Figure 1c shows the left infinite CVIS $\langle -\infty, 4, 6, \mathtt{nil} \rangle$.

We will often need to obtain the earliest and latest points in a CVIS independently of whether that point is determinate or not.

Definition 8. We define the functions *upper* and *lower* as $\mathrm{upper}(V) = \max_{\mathrm{vp}}(V.D_e, V.I_e)$ and $\mathrm{lower}(V) = \min_{\mathrm{vp}}(V.D_s, V.I_s)$, both with signature $\mathcal{CV} \rightarrow \mathcal{P}$.

We will also need to know the latest point at which a CVIS V can start and the earliest point at which it can end. To understand these functions, observe that if a CVIS V has a determinate region then the interval will start no later than

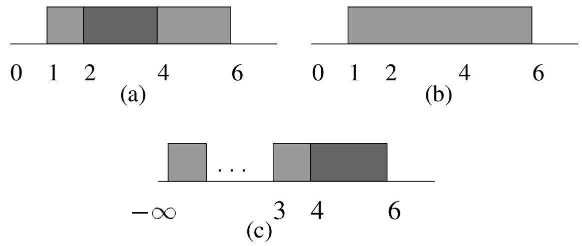

Figure 1: Examples of convex valid interval stamps

$V.D_s$ and end no earlier than $V.D_e$. On the other hand, if V is fully indeterminate then it may start as late as $V.I_e$ and end as earlier as $V.I_s$.

Definition 9. We define two functions, *maxlo* and *minup*, with signature $\mathcal{CV} \rightarrow \mathcal{P}$ as $\mathrm{maxlo}(V) = \min_{\mathrm{vp}}(V.D_s, V.I_e)$ and $\mathrm{minup}(V) = \max_{\mathrm{vp}}(V.I_s, V.D_e)$.

The following extends the definition of membership in an interval.

Definition 10. We denote membership of a point p in a CVIS V as $p \in V$ and define that membership as $p \in V \Leftrightarrow \mathrm{lower}(V) \leq p \leq \mathrm{upper}(V)$. We will also use $p \in V.D$ to represent p's determinate inclusion in V and $p \in V.I$ to represent p's indeterminate inclusion in V.

Next we extend the concept of intersection to convex valid interval stamps.

Definition 11. Two CVISs, V_1 and V_2, *intersect* if: $\mathrm{lower}(V_1) \leq \mathrm{upper}(V_2) \wedge \mathrm{lower}(V_2) \leq \mathrm{upper}(V_1)$. We say V_1 and V_2 *determinately intersect* if $V_1.D \neq \emptyset \wedge V_2.D \neq \emptyset \wedge V_1.D\ \mathit{intersects}\ V_2.D$.

The concept of adjacency can be extended to convex valid interval stamps as follows.

Definition 12. We say that two CVISs, V_1 and V_2, are *adjacent* if $\mathrm{next}(\mathrm{upper}(V_1)) = \mathrm{lower}(V_2) \vee \mathrm{next}(\mathrm{upper}(V_2)) = \mathrm{lower}(V_1)$. We say that V_1 and V_2 are *determinately adjacent* if $V_1.D \neq \emptyset \wedge V_2.D \neq \emptyset \wedge V_1.D\ \mathit{adjacent}\ V_2.D$. Note that two CVISs which are determinately adjacent may intersect but will not determinately intersect.

The work of (Cowley 1999) depicts how to translate from interval constraints of the form $\mathcal{R}\ I$, where $\mathcal{R}$ is an interval relationship and I is a determinate interval, to a convex valid interval stamp.

We will also need to consider non-convex VISs.

Definition 13. A *non-convex valid interval stamp* (NVIS) is a finite set of CVISs which meet two conditions. 1) No two CVISs may intersect. 2) If any two CVISs are adjacent, then both must have a definite interval and they must not be determinately adjacent. $\mathcal{NV}$ refers to the set of all NVISs. A CVIS can be implicitly converted to an NVIS with the function *nvis*: $\mathcal{CV} \rightarrow \mathcal{NV}$, defined as $\text{nvis}(C) = \{C\}$.

Definition 14. $\mathcal{V} = \mathcal{CV} \cup \mathcal{NV}$ is the set of all VISs.

Operators on Valid Interval Stamps There are a number of operators on VISs which correspond to the similarly named ones on Boolean expressions and sets. Specifically, we will define conjunction, intersection, union, and difference of VISs below. To avoid confusion, we will annotate the usual operators to emphasize that they carry different semantics than the familiar ones. For example, we use $\overset{v}{\cup}$ for the union of VISs. When operands are known to be convex (non-convex) VISs, we will use $\overset{cv}{\cup}$ ($\overset{nv}{\cup}$). When the familiar semantics from logic or set theory are sufficient, we will use the unannotated operators.

Conjunction There are times when a single interval constraint may not adequately express the knowledge one has about a fact's valid time. For that we must conjoin multiple constraints. We will do this by way of the VIS conjunction operator[1]: $\overset{v}{\wedge}$. The effect of $\overset{v}{\wedge}$ is to produce a determinate region which includes the determinate region of both operands and an indeterminate region which includes only points in the indeterminate region of both operands. The operator uses the knowledge expressed by two VISs in order to increase the accuracy to which the valid time of a fact is known.

In order to compute the resulting VIS, V_r, from the conjunction of two existing CVISs, V_1 and V_2, we must first ensure that the conjunction is satisfiable. For example, there are no dates which satisfy the condition "contains Jan-98 and during Mar-98". On the other hand, "contains Jan-98 and after Nov-97" can be satisfied. Each operand restricts the range of the other.

Definition 15. Two CVISs, V_1 and V_2, are *conjunction compatible*, denoted $V_1 \wedge_{\text{comp}} V_2$, if 1) V_1 intersects V_2, 2) $V_2.D \neq \emptyset \Rightarrow (\text{lower}(V_1) <= V_2.D_s \wedge V_2.D_e <= \text{upper}(V_1))$, and 3) $V_1.D \neq \emptyset \Rightarrow (\text{lower}(V_2) <= V_1.D_s \wedge V_1.D_e <= \text{upper}(V_2))$. We extend this definition to NVISs by saying that two NVISs are *conjunction compatible* if $\forall V_{1_i} \in V_1 \, \exists V_{2_j} \in V_2$ such that $V_{1_i} \wedge_{\text{comp}} V_{2_j}$ and $\forall V_{2_j} \in V_2 \, \exists V_{1_i} \in V_1$ such that $V_{2_j} \wedge_{\text{comp}} V_{1_i}$.

Conjunction compatibility ensures that the indefinite region of each interval encloses the definite region of the other. Otherwise, there would be a point which is in the definite region of one operand but not in the other operand at all. That is the condition which results from inconsistent interval constraints.

Definition 16 ($\overset{v}{\wedge}$). If V_1 and V_2 are VISs such that $V_1 \wedge_{\text{comp}} V_2$ we define their conjunction, denoted by $V_1 \overset{v}{\wedge} V_2$, as the VIS V_r such that $V_r.D = \{p | p \in V_1.D \vee p \in V_2.D\}$ and $V_r.I = \{p | p \in V_1.I \wedge p \in V_2.I\}$. If $\neg(V_1 \wedge_{\text{comp}} V_2)$ then $V_1 \overset{v}{\wedge} V_2 = \emptyset$.

Because the order of the interval constraints in a valid time specification should not matter, it is important that the order of VISs with respect to $\overset{v}{\wedge}$ not matter. Hence, this theorem, which is straightforward to show.

Theorem 1. $(\mathcal{V}, \overset{v}{\wedge})$ *forms a commutative monoid with identity* $V_{I\overset{v}{\wedge}} = \langle -\infty, \texttt{nil}, \texttt{nil}, \infty \rangle$.

Intersection The intersection of two CVISs is defined similarly to conjunction. The precondition is not as restrictive. It is simply necessary that the intervals intersect by definition 11. On the other hand, the result of intersection is more restrictive than that of conjunction. Specifically, the determinate region produced by conjunction includes the determinate regions of both operands. If a point is determinate in only one of the operands, it is determinate in the result. For intersection, the determinate region includes only those points determinately in *both* operands. If a point is determinately in one operand but indeterminately in the other, the intersection contains that point indeterminately.

Definition 17 ($\overset{v}{\cap}$). We define the intersection of two VISs, V_1 and V_2, denoted by $V_1 \overset{v}{\cap} V_2$, as the VIS V_r such that $V_r.D = \{p | p \in V_1.D \wedge p \in V_2.D\}$ and $V_r.I = \{p | p \in V_1 \wedge p \in V_2 \wedge (p \in V_1.I \vee p \in V_2.I)\}$.

As with $\overset{v}{\wedge}$, the following is straightforward to prove.

Theorem 2. $(\mathcal{V}, \overset{v}{\cap})$ *forms a commutative monoid with identity* $V_{I\overset{v}{\cap}} = \langle \texttt{nil}, -\infty, \infty, \texttt{nil} \rangle$

Union When an insert operation is performed for a fact whose non-temporal attributes are the same as an existing fact one of two things must happen. Either the operation is treated as a new insert, resulting in a pair of facts which differ only in their valid interval stamps; or the VIS of the existing fact is updated to include the valid time specified for the insert. We provide the union operator to support the latter semantics.

Definition 18 ($\overset{v}{\cup}$). We define the union of two VISs, V_1 and V_2, denoted by $V_1 \overset{v}{\cup} V_2$, as the VIS V_r such that $V_r.D = \{p | p \in V_1.D \vee p \in V_2.D\}$ and $V_r.I = \{p | p \notin V_r.D \wedge (p \in V_1.I \vee p \in V_2.I)\}$.

Again, the following is straightforward.

Theorem 3. $(\mathcal{V}, \overset{v}{\cup})$ *forms a commutative monoid with identity* $V_{I\overset{v}{\cup}} = \langle nil, nil, nil, nil \rangle$.

Difference When performing a deletion or update operation where the valid time specified for the operation is only a

[1] The term *conjunction* refers to the operation on interval constraints which $\overset{v}{\wedge}$ supports. It might be more appropriate to think of the effect on a VIS as *strengthens*.

portion of the valid time for the affected facts, we will need to produce the difference of two valid interval stamps. In general, such a difference results in a non-convex interval.

Definition 19 ($\overset{v}{-}$). We define the difference of two VISs, V_1 and V_2, denoted by $V_1 \overset{v}{-} V_2$ as the VIS V_r such that $V_r.D = \{p | p \in V_1.D \wedge p \notin V_2\}$ and $V_r.I = \{p | (p \in V_1.D \wedge p \in V_2.I) \vee (p \in V_1.I \wedge p \notin V_2.D)\}$.

Note that $\overset{v}{-}$ is neither commutative nor associative. The identity element, denoted by $V_{I\overset{v}{-}}$, is the empty interval.

Complexity of VIS Operators The basic operators on CVISs: $\wedge_{\text{comp}}$, $\overset{cv}{\wedge}$, $\overset{cv}{\cap}$, $\overset{cv}{\cup}$, and $\overset{cv}{-}$, are all constant time. This should be clear from their definitions, which depend solely on the values of the end points. For an NVIS V, we will use $|V|$ to indicate the number of CVISs $V_i \in V$. Based on an analysis of the algorithms developed to implement the operators, we have the results in table 1 for NVISs. Those algorithms and proofs of these results are omitted due to lack of space.

Operator	Complexity
$\wedge_{\text{comp}}$	$O(\lvert V_1\rvert \lvert V_2\rvert)$
$\overset{nv}{\wedge}$	$O(\lvert V_1\rvert \lvert V_2\rvert)$
$\overset{nv}{\cap}$	$O(\lvert V_1\rvert \lvert V_2\rvert)$
$\overset{nv}{\cup}$	$O((\lvert V_1\rvert \lvert V_2\rvert)^2)$
$\overset{nv}{-}$	$O(\lvert V_2\rvert^4 \lvert V_1\rvert)$

Table 1: Complexity of Operators over NVISs

Indeterminate Interval Relationships

Now we turn to comparing the indeterminate intervals by examining how the thirteen interval relationships used in Allen's interval algebra (Allen 1983) can be extended to the indeterminate intervals represented by CVISs. We will need to define both definite and potential satisfaction of the relationships when applied to CVISs, resulting in a total of 26 interval relationships. Two principles are used in reaching these definitions.

Principle 1. For V_1 definitely $\mathcal{R}$ V_2 to hold, where $\mathcal{R}$ is an interval relationship, we must have:

$$\nexists V_1', V_2' \text{ such that} V_1 \wedge_{\text{comp}} V_1' \wedge V_2 \wedge_{\text{comp}} V_2' \wedge \\ \neg((V_1 \overset{cv}{\wedge} V_1').D\, \mathcal{R}\, (V_2 \overset{cv}{\wedge} V_2').D)$$

If this principle is met, it is not possible to conjoin any compatible interval constraint to either V_1 or V_2 such that $\mathcal{R}$ will not be satisfied.

Principle 2. For V_1 potentially $\mathcal{R}$ V_2 to hold, where $\mathcal{R}$ is an interval relationship, we must have:

$$\exists V_1', V_2' \text{ such that} V_1 \wedge_{\text{comp}} V_1' \wedge V_2 \wedge_{\text{comp}} V_2' \wedge \\ (V_1 \overset{cv}{\wedge} V_1') \text{ definitely } \mathcal{R}\, (V_2 \overset{cv}{\wedge} V_2')$$

This ensures that it is possible, via the conjunction of additional interval constraints with V_1 and V_2, to arrive at a pair of intervals which definitely satisfy $\mathcal{R}$.

Definition 20. For two CVISs, V_1 and V_2, we say:

- V_1 definitely equals V_2 if
$$V_1.I = \emptyset \wedge V_2.I = \emptyset \wedge V_1.D = V_2.D$$
- V_1 potentially equals V_2 if
$$\text{lower}(V_1) \le \text{maxlo}(V_2) \wedge \text{lower}(V_2) \le \text{maxlo}(V_1) \wedge \\ \text{minup}(V_2) \le \text{upper}(V_1) \wedge \text{minup}(V_1) \le \text{upper}(V_2)$$
- V_1 definitely before V_2 if
$$\text{upper}(V_1) < \text{lower}(V_2)$$
- V_1 potentially before V_2 if
$$\text{minup}(V_1) < \text{maxlo}(V_2)$$
- V_1 definitely meets V_2 if
$$V_1.I_e = \texttt{nil} \wedge V_2.I_s = \texttt{nil} \wedge V_1.D_e = V_2.D_s \wedge \\ V_1.D_s < V_2.D_s \wedge V_1.D_e < V_2.D_e$$
- V_1 potentially meets V_2 if
$$\text{lower}(V_1) < \text{maxlo}(V_2) \wedge \text{minup}(V_1) < \text{upper}(V_2) \wedge \\ \text{lower}(V_2) \le \text{upper}(V_1) \wedge \text{minup}(V_1) \le \text{maxlo}(V_2)$$
- V_1 definitely overlaps V_2 if
$$V_1.D_s < \text{lower}(V_2) \wedge \text{upper}(V_1) < V_2.D_e \wedge \\ V_2.D_s < V_1.D_e$$
- V_1 potentially overlaps V_2 if
$$\text{lower}(V_1) < \text{maxlo}(V_2) \wedge \text{minup}(V_1) < \text{upper}(V_2) \wedge \\ \text{lower}(V_2) < \text{upper}(V_1) \wedge \\ \text{lower}(V_1) < \text{previous}(\text{upper}(V_1)) \wedge \\ \text{lower}(V_2) < \text{previous}(\text{upper}(V_2))$$
- V_1 definitely during V_2 if
$$V_2.D_s < \text{lower}(V_1) \wedge \text{upper}(V_1) < V_2.D_e$$
- V_1 potentially during V_2 if
$$\text{lower}(V_2) < \text{maxlo}(V_1) \wedge \text{minup}(V_1) < \text{upper}(V_2) \wedge \\ \text{lower}(V_2) < \text{previous}(\text{upper}(V_2))$$
- V_1 definitely starts V_2 if
$$V_1.I_s = \texttt{nil} \wedge V_2.I_s = \texttt{nil} \wedge V_1.D_s = V_2.D_s \wedge \\ \text{upper}(V_1) < \text{minup}(V_2)$$
- V_1 potentially starts V_2 if
$$\text{lower}(V_1) \le \text{maxlo}(V_2) \wedge \text{lower}(V_2) \le \text{maxlo}(V_1) \wedge \\ \text{minup}(V_1) < \text{upper}(V_2)$$
- V_1 definitely finishes V_2 if
$$V_1.I_e = \texttt{nil} \wedge V_2.I_e = \texttt{nil} \wedge V_1.D_e = V_2.D_e \wedge \\ V_2.D_s < \text{lower}(V_1)$$

- V_1 potentially finishes V_2 if

$$\mathrm{minup}(V_2) \leq \mathrm{upper}(V_1) \wedge \mathrm{minup}(V_1) \leq \mathrm{upper}(V_2) \wedge \mathrm{lower}(V_2) < \mathrm{maxlo}(V_1)$$

We can make some interesting observations about the compatibility of the various interval relationships. First, we note that the interval relationships between determinate intervals are mutually exclusive.

Lemma 1. *Let I_1, I_2 be intervals and $\mathcal{R}_1$, $\mathcal{R}_2$ be determinate interval relationships. Then, $I_1 \, \mathcal{R}_1 \, I_2 \Rightarrow \neg(I_1 \, \mathcal{R}_2 \, I_2)$.*

Proof. This follows directly from the definitions of the relationships. □

This observation extends to the definite relationships between valid interval stamps.

Theorem 4. *Let V_1 and V_2 be valid interval stamps and let $\mathcal{R}_1$ and $\mathcal{R}_2$ be determinate interval relationships, $\mathcal{R}_1 \neq \mathcal{R}_2$. Then, $V_1 \,\mathrm{definitely}\, \mathcal{R}_1 \, V_2 \Rightarrow \neg(V_1 \,\mathrm{definitely}\, \mathcal{R}_2 \, V_2)$.*

Proof. Assume that $V_1 \,\mathrm{definitely}\, \mathcal{R}_1 \, V_2$. Then $\exists V_1', V_2'$ such that $(V_1 \stackrel{cv}{\wedge} V_1').D \, \mathcal{R}_1 \, (V_2 \stackrel{cv}{\wedge} V_2').D$. By lemma 1, we know that $\neg((V_1 \stackrel{cv}{\wedge} V_1').D \, \mathcal{R}_2 \, (V_2 \stackrel{cv}{\wedge} V_2').D)$. Principle 1 then requires that $\neg(V_1 \,\mathrm{definitely}\, \mathcal{R}_2 \, V_2)$. □

We cannot make the same conclusion regarding potential relationships, however.

Theorem 5. *Let V_1 and V_2 be valid interval stamps and let $\mathcal{R}_1$ and $\mathcal{R}_2$ be determinate interval relationships, $\mathcal{R}_1 \neq \mathcal{R}_2$. Then, $V_1 \,\mathrm{potentially}\, \mathcal{R}_1 \, V_2 \not\Rightarrow \neg(V_1 \,\mathrm{potentially}\, \mathcal{R}_2 \, V_2)$.*

Proof. By example: Let p_1 and p_2 be points, $p_1 < p_2$. Let $V_1 = V_2 = \langle p_1, \mathtt{nil}, \mathtt{nil}, p_2 \rangle$. Choose $V_1' = V_2' = \langle \mathtt{nil}, p_1, p_2, \mathtt{nil} \rangle$. Then, $(V_1 \stackrel{cv}{\wedge} V_1').D \,\mathrm{equals}\, (V_2 \stackrel{cv}{\wedge} V_2').D$. By principle 2 $V_1 \,\mathrm{potentially\ equals}\, V_2$. Now, choose $V_1'' = \langle \mathtt{nil}, p_1, p_1, \mathtt{nil} \rangle$ and $V_2'' = \langle \mathtt{nil}, p_2, p_2, \mathtt{nil} \rangle$. So, $(V_1 \stackrel{cv}{\wedge} V_1'').D \,\mathrm{before}\, (V_2 \stackrel{cv}{\wedge} V_2'').D$ and $V_1 \,\mathrm{potentially\ before}\, V_2$. □

Finally, we can see that, as expected, a definite relationship implies a potential one.

Theorem 6. *Let V_1 and V_2 be valid interval stamps and let $\mathcal{R}$ be a determinate interval relationship. Then, $V_1 \,\mathrm{definitely}\, \mathcal{R} \, V_2 \Rightarrow V_1 \,\mathrm{potentially}\, \mathcal{R} \, V_2$.*

Proof. Assume $V_1 \,\mathrm{definitely}\, \mathcal{R} \, V_2$. Choose $V_1' = V_2' = V_{I\stackrel{v}{\wedge}}$. Then $V_1 \stackrel{cv}{\wedge} V_1' = V_1$ and $V_2 \stackrel{cv}{\wedge} V_2' = V_2$, so $(V_1 \stackrel{cv}{\wedge} V_1') \,\mathrm{definitely}\, \mathcal{R} \, (V_2 \stackrel{cv}{\wedge} V_2')$. This meets the conditions for principle 2. Therefore, $V_1 \,\mathrm{potentially}\, \mathcal{R} \, V_2$. □

Conclusions and Future Work

We have shown a representation for both convex and non-convex indeterminate intervals and provided a useful set of operators on those intervals. Further, we have shown that it is possible to extend the relationships used in Allen's interval algebra (Allen 1983) to the indeterminate intervals represented by convex valid interval stamps. These extensions are interesting in their own right because of the large amount of prior work in both the artificial intelligence and database fields that is based on the interval algebra and its variants. One specific application area is in allowing temporal integrity constraints to take into account indeterminism.

An obvious extension of this current work would be to reformulate Allen's constraint propagation algorithms to use indeterminate intervals and investigate resulting differences in complexity and reasoning power. It would also be interesting to reformulate the potential and definite relationships presented here using intervals treated as primitive constructs instead of based on end points. For determinate intervals, all thirteen relationships can be defined in terms of either meets or before. This would provide the ability to use dense or continuous timelines as well as discrete.

Acknowledgements The authors would like to thank Dr. Manolis Koubarakis and the anonymous referees for their comments.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843.

Anger, F. D.; Mata-Toledo, R. A.; Morris, R. A.; and Rodriguez, R. V. 1988. A relational knowledge base with temporal reasoning. In *Proceedings of the First Florida Artificial Intelligence Research Symposium*, 147–151.

Benthem, J. v. 1991. *The Logic of Time*. Dordrecht, Holland: Kluwer Academic Publishers.

Bouzid, M., and Mouaddib, A.-I. 1998. Uncertain temporal reasoning for the distributed transportation scheduling problem. In *Proceedings : Fifth International Workshop on Temporal Representation and Reasoning*, 21–28.

Cowley, W. 1999. Temporal integrity constraints with temporal indeterminacy. Master's thesis, University of South Florida.

Dyreson, C. E., and Snodgrass, R. T. 1998. Supporting valid-time indeterminacy. *Transactions on Database Systems* 23(1):1–57.

Gadia, S. K.; Nair, S. S.; and Poon, Y.-C. 1992. Incomplete information in relational temporal databases. In *Proceedings of the 18th International Conference on Very Large Data Bases*, 395–406.

Koubarakis, M. 1994. Database models for infinite and indefinite temporal information. *Information Systems* 19(2):141–174.

Koubarakis, M. 1997. The complexity of query evaluation in indefinite temporal constraint databases. *Theoretical Computer Science* 171(1–2):25–60.

Plexousakis, D. 1995. Compilation and simplification of temporal integrity constraints. In *Rules in Database Systems: Second International Workshop*, 260–274.