Going Far, Logically

Fusun Yaman and **Dana Nau** and **V.S. Subrahmanian**
Department of Computer Science,
University of Maryland Institute for Advanced Computer Studies (UMIACS),
and Institute for Systems Research (ISR)
University of Maryland
College Park, MD, 20742
{fusun, nau, vs}@cs.umd.edu

Abstract

There are numerous applications where we need to ensure that multiple moving objects are sufficiently far apart. Furthermore, in many moving object domains, there is positional indeterminacy — we are not 100% sure exactly when a given moving object will be at a given location. [Yaman *et al.*, 2004] provided a logic of motion but did not provide algorithms to ensure that moving objects are kept sufficiently far apart. In this paper, we extend their logic to include a "far" predicate. We develop the **CheckFar** algorithm that checks if any given two objects will always be sufficiently far apart at during a time interval. We have run a set of experiments showing that our **CheckFar** algorithm scales very well.

1 Introduction

All of us fly in airplanes. We know that flight plans are almost never 100% accurate. We all have a vested interest in ensuring that planes we fly in are sufficiently far from other planes that are flying in the sky at the same time. The goal of this paper is to develop a hybrid *logical and constraint based framework* to ensure that moving objects with *positional uncertainty* (where will the object be at a given time) remain sufficiently far apart.

Our work builds on [Yaman *et al.*, 2004] who developed the concept of a "go-theory". A go-theory is a finite set of "go-atoms." A go-atom can express statements such as '*Plane p1 leaves location L1 some time between 10 and 12 and arrives at location L2 at some time between 30 and 40 and during the flight the speed of the plane is between 5 and 6*". They provide algorithms to check if a given plane is within a given region at a given time point. They also introduce ground atoms of the form $\mathsf{near}(o_1, o_2, d, t_1, t_2)$ — intuitively, this means that at all times during the interval t_1, t_2, o_1 and o_2 are guaranteed to be within d units of each other.

In this paper, we introduce a predicate symbol called $\mathsf{far}(o_1, o_2, d, t_1, t_2)$ — intuitively, this means that at all times during the interval t_1, t_2, o_1 and o_2 are guaranteed to be at least d units apart.

One may think that $\mathsf{near}()$ and $\mathsf{far}()$ are complements of each other. Unfortunately, this is not true. Yaman et. al.[Yaman *et al.*, 2004] define interpretations and a notion of satisfaction. For $\mathsf{near}(o_1, o_2, d, t_1, t_2)$ atom to be a logical consequence of a go-theory G, for every interpretation $\mathcal{I}$ that satisfies G, it must be the case that for all times $t_1 \leq t \leq t_2$, the distance between the locations of o_1 and o_2 at time t is less than or equal to d. Thus, $\neg\mathsf{near}(()o_1, o_2, d, t_1, t_2)$ is a logical consequence of G if for every interpretation $\mathcal{I}$ which satisfies G *there exists* a time $t_1 \leq t \leq t_2$, the distance between the locations of o_1 and o_2 at time t (according to the interpretation $\mathcal{I}$) is greater than d. In contrast, for $\mathsf{far}(o_1, o_2, d, t_1, t_2)$ to be entailed by S, it must be the case that for every interpretation $\mathcal{I}$ which satisfies G and for *every time point* $t_1 \leq t \leq t_2$, the distance between the locations of o_1 and o_2 at time t is greater than d. Thus, entailment of $\mathsf{far}()$ atoms is not the same as entailment of either $\mathsf{near}()$ atoms or $\neg\mathsf{near}()$ literals.

In this paper, we define the semantics of $\mathsf{far}()$ and develop an algorithm called **CheckFar** to check entailment of ground $\mathsf{far}()$ atoms by a go-theory G. We have conducted extensive experiments on the computational feasibility of **CheckFar** — our experiments show in a compelling way that in real world situations[1], **CheckFar** will work very well indeed. In our experiments we answered far queries in less than 0.6 seconds for go theories up to 1000 atoms per object.

2 Background On go-Theories

We now provide a quick overview of go-theories from[Yaman *et al.*, 2004]. We assume the existence of several sets of constant symbols: $\mathbf{R}$ is the set of all real numbers, $\mathbf{O}$ is the set of names of objects, $\mathbf{P} = \mathbf{R} \times \mathbf{R}$ is the set of all points in two-dimensional cartesian space. We assume the existence of three disjoint sets of variable symbols, $V_{\mathbf{R}}$, $V_{\mathbf{O}}$, and $V_{\mathbf{P}}$, ranging over $\mathbf{R}$, $\mathbf{O}$ and $\mathbf{P}$, respectively. A *real* term t is any member of $\mathbf{R} \cup V_{\mathbf{R}}$. Object terms and point terms are defined similarly. Ground terms are defined in the usual way. We now define atoms as follows.

- If o_1, o_2 are object terms, and d, t_1, t_2 are positive real terms, then $\mathsf{near}(o_1, o_2, d, t_1, t_2)$ is an *atom*. Intuitively, this atom says that o_1, o_2 are within distance d of each other during the time interval $[t_1, t_2]$.
- If o is an object term, P_1, P_2 are point terms, and t_1, t_2 are positive real terms, then $\mathsf{in}(o, P_1, P_2, t_1, t_2)$ is an

[1] We built an application to manage separation between ships in port shipping lanes.

atom. Intuitively, this atom says that object o is in the rectangle whose lower left (resp. upper right) corner is P_1 (resp. P_2) at some point in the time interval $[t_1, t_2]$.

- If o is an object term, P_1, P_2 are point terms, and $t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+$ are positive real terms, then $\mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ is an atom called a *go atom*. Intuitively, this atom says that object o leaves point P_1 at some time in $[t_1^-, t_1^+]$ and arrives at point P_2 during $[t_2^-, t_2^+]$, traveling in a straight line with a minimum speed v^- and maximum speed v^+.

Ground atoms are defined in the usual way. A *go theory* is a finite set of *ground* go-atoms. *Note that go-theories do not contain* near() *or* in() *atoms.*

Notation. If $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$, then,

$$\begin{array}{lll} obj(g) = o, & v^-(g) = v^-, & v^+(g) = v^+, \\ loc_1(g) = P_1, & t_1^-(g) = t_1^-, & t_1^+(g) = t_1^+, \\ loc_2(g) = P_2, & t_2^-(g) = t_2^-, & t_2^+(g) = t_2^+. \end{array}$$

If G is a go-theory and o is an object id, the *restriction of* G *to* o*, denoted* G^o is the set $\{g \in G \mid obj(g) = o\}$.

An *interpretation* is a continuous function $\mathcal{I} : \mathbf{O} \times \mathbf{R}^+ \rightarrow \mathbf{P}$. Intuitively, $\mathcal{I}(o, t)$ is o's location at time t.

Definition 1 *Let* $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *be a ground atom and* $\mathcal{I}$ *be an interpretation.* $\mathcal{I}$ ***satisfies*** g *w.r.t. a time interval* $T = [t_1, t_2]$ *iff:*

- $t_1 \in [t_1^-, t_1^+]$ *and* $\mathcal{I}(o, t_1) = P_1$
- $t_2 \in [t_2^-, t_2^+]$ *and* $\mathcal{I}(o, t_2) = P_2$
- $\forall t \in [t_1, t_2]$, $\mathcal{I}(o, t)$ *is on the line segment* $[P_1, P_2]$
- $\forall t, t' \in [t_1, t_2]$, $t < t'$ *implies* $\mathsf{dist}(\mathcal{I}(o, t), P_1) < \mathsf{dist}(\mathcal{I}(o, t'), P_1)$*, where* dist *is the function that computes the Euclidean distance between two points.*
- *For all but finitely many times in* $[t_1, t_2]$, $v = d(|\mathcal{I}(o, t)|)/dt$ *is defined and* $v^-(g) \leq v \leq v^+(g)$.

This intuitively says that $I \models g$ w.r.t. a time interval $T = [t_1, t_2]$ iff o starts moving at t_1 and stops moving at t_2 and furthermore, during $[t_1, t_2]$, the object moves away from P_1 towards P_2 without either stopping or turning back or wandering away from the straight line connecting P_1 and P_2. We are now ready to define the full concept of satisfaction.

$\mathcal{I}$ *satisfies* a ground literal (denoted $\mathcal{I} \models A$) in these cases:

1. $\mathcal{I} \models \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ iff there exists an interval $[t_1, t_2]$ such that $I \models A$ w.r.t. $[t_1, t_2]$.
2. $\mathcal{I} \models \mathsf{near}(o_1, o_2, d, t_1, t_2)$ iff $\mathsf{dist}(\mathcal{I}(o_1, t), \mathcal{I}(o_2, t)) \leq d$ for all $t_1 \leq t \leq t_2$.
3. $\mathcal{I} \models \mathsf{in}(o, P_1, P_2, t_1, t_2)$ iff there are numbers $t \in [t_1, t_2]$, $x \in [{P_1}^x, {P_2}^x]$ and $y \in [{P_1}^y, {P_2}^y]$ such that $\mathcal{I}(o, t) = (x, y)$.
4. $\mathcal{I} \models \neg A$ iff $\mathcal{I}$ does not satisfy A.

The above definition can be extended in the obvious way to handle quantification — in this paper, we will only consider the ground case.

$\mathcal{I}$ *satisfies (or is a model of)* a set of ground atoms MT iff $\mathcal{I}$ satisfies every $A \in$ MT. MT is *consistent* iff there is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models$ MT. L is a *logical consequence* of MT, denoted MT $\models L$, iff every model of MT is also a model of L.

3 far() atoms

We are now ready to extend the logic in [Yaman *et al.*, 2004] to include far() atoms. If o_1, o_2 are objects, and d, t_1, t_2 are real terms, then $\mathsf{far}(o_1, o_2, d, t_1, t_2)$ is a far() atom.

Definition 2 (satisfaction of far() atoms) *Suppose* f *is a ground* far *atom and* $\mathcal{I}$ *is an interpretation.* $\mathcal{I} \models f$ *iff for all* $t_1 \leq t \leq t_2$, $\mathsf{dist}(I(o_1, t), I(o_2, t)) > d$.

We say a go theory G *entails* f iff whenever $\mathcal{I} \models G$, it is also the case that $\mathcal{I} \models f$. The far()***-entailment problem*** is that of checking whether a given go-theory entails a *ground* far()-atom. The entailment of a negated near literal does not imply entailment of the associated far atom. Similarly entailment of a negated far literal does not imply entailment of the associated near atom. However, the following results hold:

Lemma 1 *Suppose* G *is a go theory and suppose* $\mathsf{far}(o_1, o_2, d, t_1, t_2)$ *and* $\mathsf{near}(o_1, o_2, d, t_1, t_2)$ *are ground. If* $G \models \mathsf{far}(o_1, o_2, d, t_1, t_2)$*, then* $G \models \neg\mathsf{near}(o_1, o_2, d, t_1, t_2)$*. If* $G \models \mathsf{near}(o_1, o_2, d, t_1, t_2)$*, then* $G \models \neg\mathsf{far}(o_1, o_2, d, t_1, t_2)$.

The following result tells us that checking for entailment of a ground far()-atom is co-NP complete.

Theorem 1 *Let* G *be a go theory and* $f = \mathsf{far}(o, o', d, t_1, t_2)$ *be a ground atom. Checking if* $G \models f$ *is coNP-complete.*

We will omit the proofs due to space constraints.

4 Far Algorithm

Finding an algorithm to solve the far()-entailment problem is a complex task. Our solution involves the following steps. First we partition a go-theory into clusters G^o of go-atoms about the same object o. Next, for any given go-atom g, we define the *temporal certainty interval* which specifies a time interval when the object is guaranteed to be on the line segment $[loc_1(g), loc_2(g)]$. Third, we define the *positional certainty interval* that finds the smallest subsegment of the above line in which the vehicle is guaranteed to be at time t. Both temporal and positional certainty intervals are defined w.r.t. a given ordering of go-atoms. Our fourth step is to explain how to solve the far()-entailment problem when a go-theory has only two go-atoms (one for each object) and $t_1 = t_2$ in the far() atom whose entailment we are trying to check. Our fifth step generalizes this to the case when $t_1 \leq t_2$. Finally, we show how to remove the assumption that G contains only two go-atoms.

Due to space constraints, we will proceed under the assumption that G is non-collinear as defined below. Extending our algorithm to remove this assumption is straightforward [2] Our implementations do not require this assumption.

[2][Yaman *et al.*, 2004] provides a technique to merge multiple collinear go-atoms into "movements" that are non-collinear. The same technique can be used here.

Definition 3 *A go theory G is* non-collinear *iff there are no go-atoms $g, g' \in G$ such that $obj(g) = obj(g')$ and the following two conditions both hold:*

- *The intersection of line segments $[loc_1(g), loc_2(g)]$ and $[loc_1(g'), loc_2(g')]$ is a line segment $[P, Q]$ such that P is visited before Q in both g and g' and*
- *$t_1^-(g) \leq t_2^+(g')$ and $t_1^-(g') \leq t_2^+(g)$.*

4.1 Temporal and Positional Certainty Intervals

In this section, we define the notions of temporal and positional uncertainty.

Definition 4 ($\mathcal{C}(G, o, \sqsubseteq)$) *Let G be a go theory and o be an object id. Let $\sqsubseteq$ be any total ordering on G^o. The set $\mathcal{C}(G, o, \sqsubseteq)$ of linear constraints associated with $G, o, \sqsubseteq$ is defined as follows:*

- $\forall g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+) \in G^o$
  - $t_1^- \leq S_g \leq t_1^+$ *and* $t_2^- \leq E_g \leq t_2^+$
  - $v^- \times (E_g - S_g) \leq \mathsf{dist}(P_1, P_2) \leq v^+ \times (E_g - S_g)$
- *for every $g, g' \in G^o$ such that $g \sqsubseteq g'$: $E_g \leq S_{g'}$.*

S_g and E_g are variables associated with an atom g which intuitively denote the start time and end time of the movement described by g. When G^o has only one atom g,we will use $\mathcal{C}(g)$ as a short hand notation.

Example 1 *Let $g = \mathsf{go}(o, (40, 10), (70, 50), 12, 13, 21, 21, 4, 10)$ and $g' = \mathsf{go}(o, (70, 50), (30, 80), 20, 21, 30, 31, 4, 10)$. Suppose $G = \{g, g'\}$ and $g \sqsubseteq g'$. Then $\mathcal{C}(G, o, \sqsubseteq)$ contains the following constraints:*

- $12 \leq S_g \leq 13, 21 \leq E_g \leq 21$,
- $4 \times (E_g - S_g) \leq 50 \leq 10 \times (E_g - S_g)$.
- $20 \leq S_{g'} \leq 21, 30 \leq E_{g'} \leq 31$,
- $4 \times (E_{g'} - S_{g'}) \leq 50 \leq 10 \times (E_g - S_g)$,
- $E_g \leq S_{g'}$.

We now use linear programming methods to define the temporal certainty interval of a go atom.

Definition 5 ($TCI(G, o, \sqsubseteq, g)$) *Let G be a go theory, o be an object, and $\sqsubseteq$ be a total order on G^o such that $\mathcal{C}(G, o, \sqsubseteq)$ has a solution. The* ***temporal certainty interval****, $TCI(G, o, \sqsubseteq, g)$, of an atom $g \in G^o$ w.r.t. $\sqsubseteq$ is the time interval $[T^-, T^+]$ where*

- *T^- is the result of solving the linear program: "**maximize** S_g **subject to** $\mathcal{C}(G, o, \sqsubseteq)$",*
- *T^+ is the result of solving the linear program: "**minimize** E_g **subject to** $\mathcal{C}(G, o, \sqsubseteq)$".*

$TCI(G, o, \sqsubseteq, g)$ is undefined if $T^- > T^+$.

When G^o has only one atom g we will $TCI(g)$ as a short hand notation. Intuitively, $TCI(G, o, \sqsubseteq, g)$ is the interval when we know for sure that object o is within the line specified in go-atom g.

Example 2 *Let g and g' be the two go atoms in Example 1. $TCI(g)$ is $[13, 21]$. If $G = \{g, g'\}$ and $g \sqsubseteq g'$, then $TCI(G, o, \sqsubseteq, g') = [21, 30]$.*

Conversely, given a time point t and a ground go-atom g, we wish to know the potential segment on the line connecting $loc_1(g)$ and $loc_2(g)$ where the object associated with g could possibly be. To find this, we need another linear program.

Definition 6 ($PCI(G, o, \sqsubseteq, g, t)$) *Let G be a go theory, o be an object, and $\sqsubseteq$ be a total order on atoms of G^o such that $\mathcal{C}(G, o, \sqsubseteq)$ has a solution. Let g be an atom in G^o such that $TCI(G, o, \sqsubseteq, g)$ is defined and t be any time point in $TCI(G, o, \sqsubseteq, g)$. The* ***positional certainty interval****, $PCI(G, o, \sqsubseteq, g, t)$, of atom g at time t w.r.t. ordering $\sqsubseteq$ is defined as follows:*

- *Let $C(g, t)$ be the set of constraints:*
  - $\frac{d_{g,t}}{v^+(g)} \leq t - S_g \leq \frac{d_{g,t}}{v^-(g)}$
  - $\frac{D - d_{g,t}}{v^+(g)} \leq E_g - t \leq \frac{D - d_{g,t}}{v^-(g)}$

  where $D = \mathsf{dist}(loc_1(g), loc_2(g))$. Note that this is a linear program when g is fixed because many terms above (such as D, $v^-(g)$,$v^+(g)$ are constants).
- *Let $d_{g,t}^{min}$ be the result of solving the linear program*

  ***minimize** $d_{g,t}$ **subject to** $\mathcal{C}(G, o, \sqsubseteq) \cup C(g, t)$.*
- *Let $d_{g,t}^{max}$ be obtained by maximizing the objective function in the above linear program (instead of minimizing).*
- *Let $P^-(G, o, \sqsubseteq, g, t)$ be the point exactly $d_{g,t}^{min}$ units away from $loc_1(g)$ on the line connecting $loc_1(g), loc_2(g)$ and likewise, let $P^+(G, o, \sqsubseteq, g, t)$ be the point exactly $d_{g,t}^{max}$ units away from $loc_1(g)$ on the same line.*

The ***positional certainty interval*** *$PCI(G, o, \sqsubseteq, g, t)$ is the line segment from $P^-(G, o, \sqsubseteq, g, t)$ to $P^+(G, o, \sqsubseteq, g, t)$.*

When G^o has only one atom g we use the shorter notation $PCI(g, t)$.

Lemma 2 *Let g be a go atom such that $TCI(g)$ is defined, $\mathcal{I}$ be an interpretation, and suppose $t \in TCI(g)$. If $\mathcal{I} \models g$, then $\mathcal{I}(o, t)$ is on the line segment $PCI(g, t)$.*

4.2 Binary go-theories about a single time point

A binary go-theory is one which contains two go-atoms $G = \{g, g'\}$ where $obj(g) = o, obj(g') = o'$. Consider a ground $\mathsf{far}()$ query $\mathsf{far}(o, o', t_1, t_2, d)$ where $t_1 = t_2$. We need to check if the distance between o and o' is guaranteed to be greater than d at time t. The following lemma presents necessary and sufficient conditions for entailment of such queries under the above assumptions.

Lemma 3 *Let $G = \{g, g'\}$ be a go-theory such that $obj(g) = o$ and $obj(g') = o'$ and let $f = \mathsf{far}(o, o', d, t, t)$ be a ground atom. $G \models f$ iff*

- *$t \in TCI(g)$ and $t \in TCI(g')$ and*
- *The minimum distance between line segments $PCI(g, t)$ and $PCI(g', t)$ is greater than d.*

Example 3 *Let $g = \mathsf{go}(o, (40, 10), (70, 50), 12, 13, 21, 21, 4, 10)$ and $g' = \mathsf{go}(o', (55, 20), (45, 80), 17, 18, 32, 33, 2, 6)$ be two atoms. Let $G = \{g, g'\}$ and $f = \mathsf{far}(o, o', 5, 19, 19)$. $TCI(g) = [13, 21]$ and $TCI(g') = [18, 32]$ both include the*

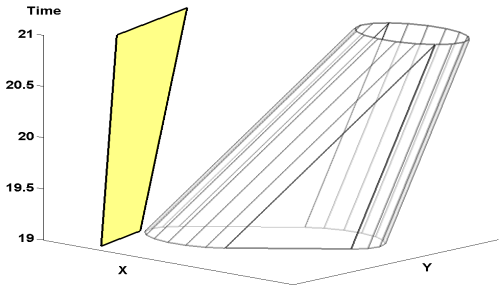

Figure 1: The polygon on the left is the space envelope of g' during $[19, 21]$ and the volume on the right is 5-neigbourhood of g during time interval $[19, 21]$. g and g' are as defined in Examples 4,5 and 6.

time point 19. At time 19, o is somewhere on the line segment $PCI(g, 19) = [(58, 34), (65.2, 43.6)]$ *and object o' is on the line segment* $PCI(g', 19) = [(54.67, 21.97), (53.08, 31.84)]$. *The minimum distance between these two lines is 33.73 which is greater than 5 so* $G \models f$.

Consider the atom $\mathsf{near}(o, o', d, t, t)$ atom instead of $\mathsf{far}(o, o', d, t, t)$ then the second bullet of Lemma 3 becomes: "*The* ***maximum*** *distance between line segments* $PCI(g, t)$ *and* $PCI(g', t)$ *is* ***less than or equal to*** d". This maximum distance is achieved at the end points of $PCI(g, t)$ and $PCI(g', t)$, e.g. $P^-(g, t)$ and $P^+(g', t)$. However the minimum distance between $PCI(g, t)$ and $PCI(g', t)$ is not necessarily at the end points hence its computation is more complex.

4.3 Binary go-theories with temporal intervals

The complexity of computing far queries gets magnified even more when we consider the case $t_1 \leq t_2$. For the near atom this is easy because it is enough to check the distance at the end points of $PCI(g, t)$ and as shown in [Yaman *et al.*, 2004] end points of $PCI(g, t)$ are piecewise linear functions over a time interval. This is not enough for the far atom. For this reason answering far queries over time intervals requires a different approach then the one in [Yaman *et al.*, 2004].

We first define the *space envelope* of a go-atom. Intuitively, the space envelope of a go-atom g is the set of all (x, y, t)-triples such that there exists a model $\mathcal{I}$ of g in which $\mathcal{I}(obj(g), t) = (x, y)$. In other words, it defines where and when it is *possible* for object o to be.

Definition 7 *Let* g *be a ground go-atom such that* $TCI(g)$ *is defined and let* $T = [t_1, t_2]$ *be any time interval such that* $T \subseteq TCI(g)$. *The* ***space envelope***, $SE(g, T)$ *of* g ***during*** *interval* T *is* $\{(x, y, t) \mid t \in T$ *and* $(x, y) \in PCI(g, t)\}$.

Theorem 2 $SE(g, T)$ *is a convex set.*

Example 4 *Let* $g' = \mathsf{go}(o', (55, 20), (45, 80), 17, 18, 32, 33, 2, 6)$ *be a go atom. The space envelope of* g' *over time interval* $[19, 21]$, *is shown in Figure 1 on the left side. It is easy to see that* $SE(g', [19, 21])$ *is convex.*

We now define the set of points that are closer than a given distance d to any possible location of an object at a given time.

Definition 8 (d-neighbourhood) *Let* g *be a ground go-atom such that* $TCI(g)$ *is defined. Let d be a real number and let* $T = [t_1, t_2]$ *be a time interval such that* $T \subseteq TCI(g)$. *The* ***d-neighbourhood*** *of* g *during* T, *denoted* $Nbr(g, T, d) = \{(x, y, t) \mid t_1 \leq t \leq t_2$ *and* $(x, y) \in NearPts(g, t, d)\}$ *where* $NearPts(g, t, d) = \{(x, y) \mid \exists (x', y') \in PCI(g, t)$ *and* $\mathsf{dist}((x, y), (x'y')) \leq d\}$.

Intuitively $NearPts(g, t, d)$ is the set of all points p such that all points on the line segment $PCI(g, t)$ which are d units or less in distance from p at time t. Similarly $Nbr(g, T, d)$ is the set of all points (x, y, t) such that it is possible for $obj(g)$ to be within d units of (x, y) at some time t in interval T.

Theorem 3 (i) $Nbr(g, T, d)$ *is a convex set.* (ii) *If* $d = 0$, *then* $Nbr(g, T, d) = SE(g, T)$.

Example 5 *Let* $g = \mathsf{go}(o, (40, 10), (70, 50), 12, 13, 19, 21, 4, 10)$ *be a go atom. The* 5*-neighborhood of* g *over time interval* $[19, 21]$ *is shown on the far right hand side of Figure 1. It is easy to see that it is a convex set.*

The following theorem states necessary and sufficient conditions under which $\{g, g'\} \models f$.

Theorem 4 *Let* $f = \mathsf{far}(o, o', d, t_1, t_2)$ *and* $G = \{g, g'\}$ *be a go theory where* $obj(g) = o$ *and* $obj(g') = o'$. $G \models f$ *iff*

- $[t_1, t_2] \subseteq TCI(g)$ *and* $[t_1, t_2] \subseteq TCI(g')$
- $Nbr(g, [t_1, t_2], d) \cap SE(g', [t_1, t_2]) = \emptyset$

Thus, an algorithm to solve the $\mathsf{far}()$-entailment problem only needs to check both these conditions. [Mantyla, 1988] provides polynomial algorithms to check for the intersection of two convex sets — these can be used directly to check the second condition above.

Example 6 *Let* $g = \mathsf{go}(o, (40, 10), (70, 50), 12, 13, 21, 21, 4, 10)$ *and* $g' = \mathsf{go}(o', (55, 20), (45, 80), 17, 18, 32, 33, 2, 6)$ *be two go atoms. Let* $G = \{g, g'\}$ *and* $f = \mathsf{far}(o, o', 5, 19, 21)$. *Then* $TCI(g) = [13, 21]$ *and* $TCI(g') = [18, 32]$. *Both include the time interval* $[19, 21]$. *Figure 1 shows* $Nbr(g, [19, 21], 5)$ *and* $SE(g', [19, 21])$. *It is apparent from the figure that the two do not intersect: hence* $G \models f$.

4.4 Arbitrary Go-theories with temporal intervals

We now remove the restriction that G is a binary go-theory. Doing so introduces several complications. For any single o, there may be many possible total orderings associated with G^o. Furthermore, for some theories we can predict the possible locations of an object not only in the certainty interval of an atom, but also during a time interval that spans over several atoms. The following lemma gives the necessary and sufficient conditions when this can be done.

Lemma 4 *Let* G *be a go-theory,* o *be an object, and* $\sqsubseteq$ *be a total order on* G^o *such that* $\mathcal{C}(G, o, \sqsubseteq)$ *has a solution. Suppose* $g_1 \sqsubseteq g_2 \ldots \sqsubseteq g_n$ *are the atoms of* G^o. $TCI(G, o, \sqsubseteq, g_i) \cup TCI(G, o \sqsubseteq, g_{i+1}) \ldots \cup TCI(G, o, \sqsubseteq, g_j)$ *is a* single time interval *iff for every* $1 \leq i \leq k < j \leq n$ *with* $TCI(G, o, \sqsubseteq, g_k) = [T_k^-, T_k^+]$ *the following are true*

- $T_k^+ = T_{k+1}^-$
- $loc_2(g_k) = loc_1(g_{k+1})$

Definition 9 *Let G be a go-theory, o be an object, and $\sqsubseteq$ be a total order on G^o such that $\mathcal{C}(G, o, \sqsubseteq)$ has a solution. Suppose $S \subseteq G^o$. S is **temporally relevant** to time interval $[t_1, t_2]$ is iff*

- *The atoms in S satisfy the conditions in lemma 4 and*
- *There are atoms g, g' in S such that $t_1 \in TCI(G, o, \sqsubseteq, g)$ and $t_2 \in TCI(G, o, \sqsubseteq, g')$*

Example 7 *Let $g_1 = \mathsf{go}(o, (40, 10), (70, 50), 12, 13, 21, 21, 4, 10)$ and $g_2 = \mathsf{go}(o, (70, 50), (30, 80), 20, 21, 30, 31, 4, 10)$ be go atoms. If $G = \{g_1, g_2\}$ and $g_1 \sqsubseteq g_2$ then $\mathcal{C}(G, o, \sqsubseteq)$ has a solution. Moreover $TCI(G, o, \sqsubseteq, g_1) = [13, 21]$, $TCI(G, o, \sqsubseteq, g_2) = [21, 30]$ and $TCI(G, o, \sqsubseteq, g_1) \cup TCI(G, o, \sqsubseteq, g_2)$ is a single interval. Suppose $T = [19, 30]$ then $\{g_1, g_2\}$ is temporally relevant to T.*

We now generalize definitions 7 and 8 to accommodate non-binary go-theories.

Definition 10 ($SE(G, o, \sqsubseteq, T)$) *Let G be a go-theory, o be an object, and $\sqsubseteq$ be a total order on G^o such that $\mathcal{C}(G, o, \sqsubseteq)$ has a solution. If $T = [t_1, t_2]$ is a time interval, then $SE(G, o, \sqsubseteq, T)$ is the set of all points (x, y, t) such that*

- *$t \in T$ and $t \in TCI(G, o, \sqsubseteq, g)$ for some $g \in G^o$,*
- *(x, y) is on $PCI(G, o, \sqsubseteq, g, t)$.*

$SE(G, o, \sqsubseteq, T)$ is not defined if there is a time point $t \in T$ such that for all $g \in G^o$, $t \notin TCI(G, \sqsubseteq, g)$.

Note that $SE(G, o, \sqsubseteq, T)$ is not necessarily convex when T spans over multiple go atoms. We can generalize Nbr to $Nbr(G, o, \sqsubseteq, T, d)$ in a similar manner — these are omitted due to space constraints. $Nbr(G, \sqsubseteq, T, d)$ also may not be convex. We now state a theorem describing the conditions under which a go theory G entails a $\mathsf{far}()$ atom.

Theorem 5 *Suppose G is a go-theory and o, o' are objects. The ground atom $f = \mathsf{far}(o, o', d, t_1, t_2)$ is a logical consequence of G iff for every pair of total orders $\sqsubseteq$ and $\sqsubseteq'$ on G^o and $G^{o'}$ such that $\mathcal{C}(G, o, \sqsubseteq)$ and $\mathcal{C}(G, o', \sqsubseteq')$ are solvable:*

- *$\exists S \subseteq G^o$ such that S is temporally relevant to $[t_1, t_2]$*
- *$\exists S' \subseteq G^{o'}$ such that S' is temporally relevant to $[t_1, t_2]$*
- *$Nbr(G, o, \sqsubseteq, [t_1, t_2], d) \cap SE(G, o', \sqsubseteq', [t_1, t_2]) = \emptyset$.*

Example 8 *Let g_1 and g_2 be go atoms in Example 7 and $g' = \mathsf{go}(o', (55, 20), (45, 80), 17, 18, 32, 33, 2, 6)$. Let $G = \{g_1, g_2, g'\}$ and $f = \mathsf{far}(o, o', 5, 19, 30)$. If $g_1 \sqsubseteq g_2$ and $\sqsubseteq' = \emptyset$ then Figure 2 shows $Nbr(G, o, \sqsubseteq, [19, 30], 5)$ and $SE(G, o', \sqsubseteq', [19, 30])$. The figure also shows that the sets intersect - hence $G \not\models f$.*

As $Nbr(G, o, \sqsubseteq, T, d)$ and $SE(G, o', \sqsubseteq, T)$ are not always convex, computing their intersection is tricky. We partition T into subintervals $T_1, T_2, \ldots, T_n$ (we call this a ***convex partition of*** T) such that for all $i < n$, the end point of T_i is the end point of the temporal certainty interval of some $g \in G^o \cup G^{o'}$. It is easy to verify that $n \leq |G^o| + |G^{o'}|$. For each T_i, $Nbr(G, o, \sqsubseteq, T_i, d)$ and $SE(G, o', \sqsubseteq, T_i)$ are convex and we leverage this in the $\mathsf{CheckFar}$ algorithm below.

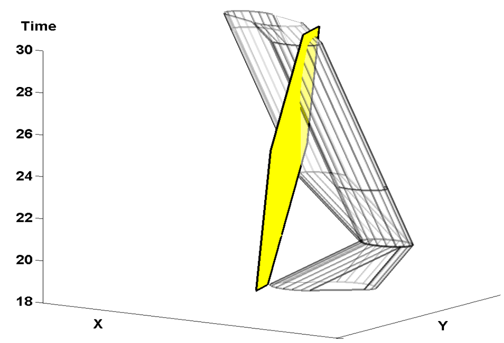

Figure 2: The space envelope of o' with respect to G and 5-neighbourhood of o w.r.t. G and $\sqsubseteq$ during time interval $[19, 30]$. G and $\sqsubseteq$ are as defined in Example 8

Algorithm CheckFar($G, \sqsubseteq, \sqsubseteq', f$)
 Suppose $f = \mathsf{far}(o, o', d, t_1, t_2)$;
 if $\mathcal{C}(G, o, \sqsubseteq)$ or $\mathcal{C}(G, o', \sqsubseteq')$ has no solution **then return** true
 Let $S_o \subseteq G^o$ such that S_o is temporally relevant to $[t_1 t_2]$
 Let $S_{o'} \subseteq G^{o'}$ such that $S_{o'}$ is temporally relevant to $[t_1 t_2]$
 if no such S_o or $S_{o'}$ exists **then return** false
 Let $T_1, T_2 \ldots T_n$ be the convex partition of $[t_1, t_2]$
 for each $i < n$ **do**
  if $Nbr(G, o, \sqsubseteq, T_i, d) \cap SE(G, o', \sqsubseteq', T_i]) \neq \emptyset$
  then return false
 end for
 return true

Theorem 6 *Suppose G is a* go*-theory and $f = \mathsf{far}(o, o', d, t_1, t_2)$ is a ground atom. Then: f is entailed by G iff for every total order $\sqsubseteq$ and $\sqsubseteq'$ on G^o and $G^{o'}$, the algorithm* CheckFar*($G, \sqsubseteq, \sqsubseteq', f$) returns "true".*

5 Implementation

We have implemented the $\mathsf{CheckFar}$ algorithm in Matlab. Our test results show that $\mathsf{CheckFar}(G, \sqsubseteq, \sqsubseteq', f)$ runs in linear time. Note that a single run of $\mathsf{CheckFar}$ does not check if $G \models f$. In fact $\mathsf{CheckFar}(G, \sqsubseteq, \sqsubseteq', f)$ should be executed for every possible $\sqsubseteq, \sqsubseteq'$ pair that produce a solvable set of constraints. Although in theory this number is exponential which leads to coNP-hardness, in practice we expect this number to be bounded by a manageable constant k. In general individual movements can have some uncertainty however the order in which they are going to be accomplished is usually well known. Figure 3 shows the running time of 3 far queries as we vary the number of atoms per object when $k = 256$. Each data point is an average of 300 runs. For these experiments we created go theories G with at most 16 total orderings and all atoms are in a rectangle of size 300 by 400. We queried those theories using three query templates $Q1 = \mathsf{far}(o, o', 1, t_1, t_2)$, $Q2 = \mathsf{far}(o, o', 50, t_1, t_2)$ and $Q3 = \mathsf{far}(o, o', 10000, t_1, t_2)$ where $[t_1, t_2]$ is a random time interval with length 10. The reader will notice that as the distance increases, the time to compute the query decreases. In the case of Q3 which is trivially false it takes almost no time to return the answer. Queries Q1 and Q2 take at most 0.6 seconds for go theories with 1000 atoms per object.

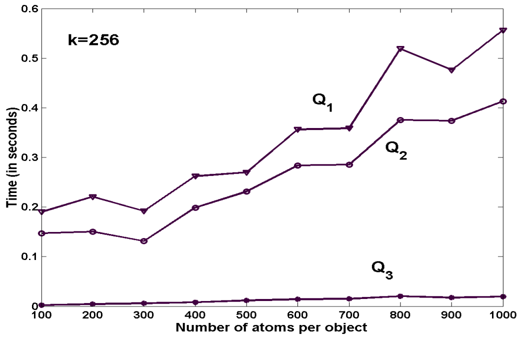

Figure 3: Time to compute three far queries Q1, Q2 and Q3 when $k = 256$.

6 Related Work

To our knowledge, there is no work on a logical foundation for checking separation between moving objects in the presence of temporal and spatial uncertainty. However, almost all models of moving entities like cars, airplanes, ships, humans and animals are subject to uncertainty about where they are at a given point in time.

[Gabelaia *et al.*, 2003; Merz *et al.*, 2003; Wolter and Zakharyaschev, ; Cohn *et al.*, 2003] study spatio-temporal logics where time is a discrete sequence rather than being continuous. [Muller,] describes a formal theory for reasoning about motion in a qualitative framework that supports complex motion classes. The work is purely symbolic. [Shanahan, 1995] discusses the frame problem, when constructing a logic-based calculus for reasoning about the movement of objects in a real-valued co-ordinate system. [Rajagopalan and Kuipers, 1994] focuses on relative position and orientation of objects with existing methods for qualitative reasoning in a Newtonian framework. The focus of these works is qualitative - in contrast our work is heavily continuous and rooted in a mix of geometry and logic rather in just logic alone. far() queries are not studied in any of these papers.

There is also some peripherally related work in computer graphics [Lin and Gottschalk, 1998] and in "moving object databases" [Erwig *et al.*, 1999] – they provide no model theory or algorithms for far() queries nor allow any uncertainty.

7 Conclusions

The ability to query a large number of objects moving uncertainly in time and space so that separation constraints between objects are maintained is critical to many applications (e.g. air traffic control, shipping port lane management). In this paper, we have extended the "logic of motion" proposed by Yaman et. al. [Yaman *et al.*, 2004] via a far() predicate. We develop a model theory for far()-entailment. We show that the far()-entailment problem is coNP-complete. We develop an efficient algorithm called **CheckFar** that effectively checks if a given total ordering of atoms in a go-theory preserve separation constraints. To check whether two objects are guaranteed to be at least d units of distance away from each other during an interval of time, we must apply **CheckFar** to different possible orderings of the atoms in the go-theory. In real-world applications such as air traffic control and a shipping port lane management we have built, we expect such orderings to be relatively small in number even if the number of objects is very large. The reason for this is that usually the flight plan for each airplane consists of a sequence of steps which only yields 1 ordering of the atoms per airplane (whereas our experiments have used 16). Our experiments show that **CheckFar** performs very well in practice.

Acknowledgements

This work was supported in part by ARO grant DAAD190310202, ARL grants DAAD190320026 and DAAL0197K0135, the ARL CTAs on Telecommunications and Advanced Decision Architectures, NSF grants IIS0329851 and IIS0412812 and 0205489 and UC Berkeley contract number SA451832441 (subcontract from DARPA's REAL program).

References

[Cohn *et al.*, 2003] A. G. Cohn, D. Magee, A. Galata, D. Hogg, and S. Hazarika. Towards an architecture for cognitive vision using qualitative spatio-temporal representations and abduction. In *Spatial Cognition III*. 2003.

[Erwig *et al.*, 1999] M. Erwig, R. H. Güting, M. Schneider, and M. Vazirgiannis. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. *GeoInformatica*, 3(3):269–296, 1999.

[Gabelaia *et al.*, 2003] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. On the computational complexity of spatio-temporal logics. In *FLAIRS Conference*, pages 460–464, 2003.

[Lin and Gottschalk, 1998] M. C. Lin and S. Gottschalk. Collision detection between geometric models: a survey. In *Proc.of IMA Conference on Mathematics of Surfaces*, 1998.

[Mantyla, 1988] M. Mantyla. *Introduction to Solid Modeling*. W. H. Freeman and Co., 1988.

[Merz *et al.*, 2003] S. Merz, J. Zappe, and M. Wirsing. A spatio-temporal logic for the specification and refinement of mobile systems. In *Fund. Appr. to Software Engr.*, pages 87–101, 2003.

[Muller,] P. Muller. A qualitative theory of motion based on spatio-temporal primitives. In *KR98*, pages 131–141.

[Rajagopalan and Kuipers, 1994] R. Rajagopalan and B. Kuipers. Qualitative spatial reasoning about objects in motion: Application to physics problem solving. In *Conf. on AI for Applications*, pages 238–245, 1994.

[Shanahan, 1995] M. Shanahan. Default reasoning about spatial occupancy. *Artificial Intelligence*, 74(1):147–163, 1995.

[Wolter and Zakharyaschev,] F. Wolter and M. Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In *KR2000*, pages 3–14.

[Yaman *et al.*, 2004] F. Yaman, D. Nau, and V. S. Subrahmanian. A logic of motion. In *KR2004*, pages 85–94, 2004.