

# Probabilistic Go Theories

Austin Parker, Fusun Yaman\*, Dana Nau and V.S. Subrahmanian

Dept. of Computer Science,  
Institute for Advanced Computer Studies,  
and Institute for Systems Research  
University of Maryland, College Park, MD 20742.  
{ austinjp,fusun,nau,vs}@cs.umd.edu

## Abstract

There are numerous cases where we need to reason about vehicles whose intentions and itineraries are not known in advance to us. For example, Coast Guard agents tracking boats don't always know where they are headed. Likewise, in drug enforcement applications, it is not always clear where drug-carrying airplanes (which do often show up on radar) are headed, and how legitimate planes with an approved flight manifest can avoid them. Likewise, traffic planners may want to understand how many vehicles will be on a given road at a given time. Past work on reasoning about vehicles (such as the "logic of motion" by Yaman et al. [Yaman et al., 2004]) only deals with vehicles whose plans are known in advance and don't capture such situations. In this paper, we develop a formal probabilistic extension of their work and show that it captures both vehicles whose itineraries are known, and those whose itineraries are not known. We show how to correctly answer certain queries against a set of statements about such vehicles. A prototype implementation shows our system to work efficiently in practice.

## 1 Introduction

There are many applications where one may wish to reason about a set of moving vehicles. One example is [Mittu and Ross, 2003] who developed (jointly with the US Navy, Lockheed Martin, BBN, and other companies) ways to predict where and when an enemy submarine would be in the future (and with what probability) based on knowledge about its past movements, terrain conditions, etc. Their predictions consist of a set of statements of the form "Vehicle  $v$  will be at location  $L$  with some probability in the interval  $[\ell, u]$ ." Likewise, BAE Systems and the US Army developed a system which makes similar predictive statements about where land vehicles will be in the future. Cell phone companies are (and in some cases already have) developed methods to predict where cell phone users

will be going in the future — a small number of law enforcement agencies in the US already use such probabilistic predictions to track selected criminals (e.g. in the case of a child abduction in the US, an *Amber alert* is used) and such predictions help determine where best to cut them off. These are three applications we know of where predictions of the form "vehicle  $v$  is going to be at location  $L$  at time  $t$  with probability  $p$ " are automatically generated and reasoned with. There are numerous theoretical models already to predict where vehicles will be in the future, when they will be there, and with what probability [Chen and Chien., 2001; Tsang et al., 1999; Kato et al., 2004]. *This paper does not reinvent the wheel by showing how to predict when and where vehicles will be in the future - this has already been done in [Chen and Chien., 2001; Tsang et al., 1999; Kato et al., 2004; Mittu and Ross, 2003] and a host of other papers. Rather, we focus on how to reason about such predictions.*

In this paper, we develop a principled approach to solving such problems by extending "go" theories due to Yaman et al. [Yaman et al., 2004; 2005]. Their framework is suitable for reasoning about applications where we know the vehicles' intended destinations — however, there are many applications such as the three mentioned above where this is not known with certainty. A second drawback of the above framework is that while temporal indeterminacy is allowed via intervals, no probability measure is associated with those intervals. This again is appropriate when we are reasoning about plans known to us (e.g. flight plans), but is not appropriate when we are reasoning about a vehicle (e.g. an enemy vehicle on the battlefield) whose plans are not known to us with 100% accuracy.

In this paper, we propose "probabilistic" go (pgo theory for short) theories by building on [Yaman et al., 2004]. A pgo theory allows us to reason about motion plans that we know as well as motion plans that we do not know with 100% certainty. The next section provides a syntax for pgo theories. The section after that gives a formal model theoretic semantics. The following section shows how to check consistency of pgo theories via linear programming. However, the size of the linear program in question may be exponential, leading one to initially suspect (wrongly) that consistency checking here is NP-complete. We subsequently determine that this problem is *polynomially solvable* (under the assumption that we are reasoning only about a finite future) by constructing

\*Fusun Yaman is currently at the University of Maryland Baltimore County (fusun@cs.umbc.edu).

a polynomially sized set of linear constraints for consistency checking and to answer certain kinds of queries called “in” queries such as “is vehicle  $id$  within a given region at a given time with probability over a threshold?” Such queries are obviously of great utility. The next section describes a prototype implementation, together with experimental results showing our system to perform well in practice.

## 2 Syntax of pgo Theories

We assume the existence of a set  $ID$  of vehicle ids. Each  $id \in ID$  has an associated maximal velocity  $v_{id}^+$ . We also assume that time is represented by integers drawn from some set  $T = [0, N]$  for some integer  $N$ . Likewise, we assume  $Space = [0, K] \times [0, K']$  is the set of all points  $(x, y)$  such that  $x, y$  are integers and  $0 \leq x \leq K, 0 \leq y \leq K'$  for some integers  $K, K'$ . We use  $ed(p_1, p_2)$  to denote the Euclidean distance between two points. We assume the existence of a set  $\mathcal{L} \subseteq Space$  called “locations.” For instance, consider a  $1024 \times 1024$  region — however, if we are only interested in reasoning about “on road” vehicles, then the only locations we might be interested in would be the locations along the roads — in this case,  $\mathcal{L}$  would consist of points on the roads.

**Definition 1 (Reachability).** We assume the existence of a reachability predicate  $reachable(id, L_1, L_2)$  which is true iff vehicle  $id$  can move to location  $L_2$  from location  $L_1$  in one unit of time. The reachability predicate must satisfy the axiom:

$$reachable(id, L_1, L_2) \Rightarrow ed(L_1, L_2) \leq v_{id}^+$$

We extend reachability to include time  $t > 0$  as follows:  $reachable(id, t, L_1, L_2)$  iff either (i)  $reachable(id, L_1, L_2)$  and  $t = 1$  or (ii) there is an  $L_3$  s.t.  $reachable(id, L_3, L_2)$  and  $reachable(id, t - 1, L_1, L_3)$ .

Intuitively, the reachability predicate encapsulates many aspects of vehicle movement that we do not wish to get into in this paper. For example, if a road is a narrow and winding road up a mountainside, the vehicle may not be able to achieve its maximal speed — in this case, the *reachable* predicate tells us what is achievable and what is not. Likewise, *reachable* also may tell us that certain locations cannot be reached by a particular vehicle, e.g. a car may not be able to drive from New York to Paris.

**Definition 2 (Atoms).** Suppose  $id, id_1, id_2 \in ID, L \in \mathcal{L}, t \in T$ , and  $p \in [0, 1]$ . Suppose  $r$  is a convex region.

1.  $pgo(id, L, t, p)$  is a probabilistic go atom (hereafter a *pgo atom*). Intuitively, this atom says that  $id$  is at location  $L$  at time  $t$  with probability  $p$ .
2.  $in(id, r, t, p)$  is a probabilistic in atom. Intuitively, this atom says that  $id$  will be somewhere in region  $r$  at time  $t$  with probability  $p$  or more.

A *pgo theory* is a finite set of *pgo atoms*. Note that in atoms *cannot* appear within a *pgo theory* but can be used in queries. As mentioned earlier, we know of at least three applications where *pgo theories* are automatically generated by prediction algorithms: the Lockheed/BBN/US Navy and other application for predicting submarine movements, the

BAE/US Army application for predicting locations of enemy vehicles, and the law enforcement application based on predicting cell phone locations.

## 3 Semantics of pgo Theories

We now define a model theory for *pgo theories*.

**Definition 3 (World).** A world  $w$  is any function from  $ID \times T$  to  $\mathcal{L}$  such that  $reachable(id, w(id, t), w(id, t + 1))$  holds for all  $id \in ID$  and  $t \in T$ .  $\mathcal{W}$  denotes the set of all worlds.

Intuitively,  $w(id, t)$  tells us where the vehicle  $id$  is at time  $t$  according to world  $w$ .

**Definition 4 (Interpretation).** An interpretation is a probability distribution  $I$  over  $\mathcal{W}$ , i.e.  $I$  assigns values in the  $[0, 1]$  interval to worlds  $w \in \mathcal{W}$  such that  $\sum_{w \in \mathcal{W}} I(w) = 1$ . We use  $\mathcal{I}$  to denote the set of all probability distributions over  $\mathcal{W}$ .

An interpretation assigns a probability to each possible world. We are now ready to define the concept of satisfaction of atoms by an interpretation.

**Definition 5 (Satisfaction).** Suppose  $I$  is an interpretation.  $I$  satisfies

1.  $pgo(id, L, t, p)$  iff  $\sum_{w \in \mathcal{W}, w(id, t) = L} I(w) = p$ .
2.  $in(id, r, t, p)$  iff  $\sum_{w \in \mathcal{W}, w(id, t) \in r} I(w) \geq p$ .

$I$  satisfies a *pgo theory*  $\mathcal{G}$  iff it satisfies all atoms in  $\mathcal{G}$ .

As usual, a *pgo theory*  $\mathcal{G}$  is consistent iff there is at least one interpretation that satisfies it. An atom  $a$  is *entailed* by  $\mathcal{G}$  iff every interpretation that satisfies  $\mathcal{G}$  also satisfies  $a$ .

For a *pgo theory*  $\mathcal{G}$ , an  $id \in ID$  and a  $t \in T$ , we let  $\mathcal{G}^{id, t}$  be the set  $\{(l_i, p_i) \mid \exists pgo(id, l_i, t, p_i) \in \mathcal{G}\}$ . For instance, if  $\mathcal{G}$  contains only  $pgo(id, (5, 5), 1, .8)$ ,  $pgo(id, (5, 6), 2, .5)$ , and  $pgo(id, (6, 5), 2, .25)$ , then  $\mathcal{G}^{id, 1} = \{(5, 5), .8\}$  and  $\mathcal{G}^{id, 2} = \{(5, 6), 0.5\}, \{(6, 5), 0.25\}$ .

**Definition 6 (Completeness).** A *pgo theory*  $\mathcal{G}$  is complete for  $id$  at  $t$  iff  $\sum_{(l, p) \in \mathcal{G}^{id, t}} p = 1$ .  $\mathcal{G}$  is complete iff it is complete for all  $id$  at all time points  $t$ .

Intuitively, a *pgo theory* is complete at time  $t$  for vehicle  $id$  iff every place that the vehicle can possibly be at at that time is explicitly mentioned in  $\mathcal{G}^{id, t}$ .

## 4 Checking Consistency of pgo Theories

In this section, we start by observing that we can check consistency of *pgo theories* by solving a set of linear constraints. For each world  $w$ , let  $v_w$  be a variable representing the probability that the world  $w$  is the actual world.

**Definition 7 (LP constraints for a pgo atom).** For *pgo atom*  $a = pgo(id, L, t, p)$ , let  $\mathbf{LP}(a)$  be the set of equations:

1.  $\sum_{w \in \mathcal{W}} v_w = 1$ ,
2. For all  $w \in \mathcal{W}, 0 \leq v_w \leq 1$ .
3.  $\sum_{w \in \mathcal{W}, w(id, t) = L} v_w = p$ ,

If  $\mathcal{G}$  is a *pgo theory*, we set  $\mathbf{LP}(\mathcal{G}) = \bigcup_{a \in \mathcal{G}} \mathbf{LP}(a)$ .

The first and second constraints force  $I$  to be a proper probability distribution. The third forces the sum of the probabilities of the worlds in which a given vehicle  $id$  is at location  $L$  at time  $t$  to be exactly  $p$  if there is a go-atom that says this. The following result give us connections between consistency of a pgo theory, and the above set of constraints.

**Proposition 1.**

- (i) A pgo theory  $\mathcal{G}$  is consistent iff  $\mathbf{LP}(\mathcal{G})$  is solvable.
- (ii) Suppose  $a = in(id, r, t, p)$ .  $\mathcal{G} \models a$  iff the result of minimizing  $\sum_{w \in \mathcal{W}, w(id, t) \in r} v_w$  subject to the constraints  $\mathbf{LP}(\mathcal{G})$  is greater than or equal to  $p$ .

An obvious problem with the above result is that the size of the input to the linear program for  $\mathbf{LP}(\mathcal{G})$  is on the order of  $|\mathcal{L}|^{|T| \cdot |ID|} \times |\mathcal{G}|$ . This is too large for the above algorithms to tractably solve any reasonably sized problem. One may wonder whether consistency checking for pgo-theories is NP-complete. It is not, as we will shortly see in the next section.

## 5 Partial Path Probabilities

$\mathbf{LP}(\mathcal{G})$  associates a variable in the linear program with each world. Instead, we might want to associate a variable  $p[id, t, L, L']$  denoting the probability that a vehicle with ID  $id$  travels from  $L$  to  $L'$  leaving at time  $t$ . We call this a *path probability variable*. It is clear that as long as we only look at a bounded time horizon, the number of path probability variables is polynomial w.r.t. the number of time points, size of *Space* and the number of vehicles. What we will try to do in this section is to reformulate  $\mathbf{LP}(\mathcal{G})$  in terms of these variables so that the resulting set of constraints is polynomial in the size of the pgo-theory.

**Definition 8** (Interpretation Compatibility). Given  $p[id, t, L, L']$  defined for every  $id, t, L, L'$  and interpretation  $I$ , we say  $I$  is compatible with  $p$  iff

$$p[id, t, L, L'] = \sum_{w(id, t) = L, w(id, t+1) = L'} I(w)$$

**Theorem 1.** Suppose  $\theta$  is an assignment to all path probability variables. There is an interpretation  $I$  compatible with  $\theta$  iff  $p$  satisfies

- 1. For each  $t, id$ ,  $\sum_{L \in \mathcal{L}} \sum_{L' \in \mathcal{L}} p_\theta[id, t, L, L'] = 1$ .
- 2. For each  $t, id, L, L'$   $p_\theta[id, t, L, L'] \geq 0$ .
- 3.  $\neg reachable(id, L, L') \rightarrow \forall t, p_\theta[id, t, L, L'] = 0$ .
- 4. For each  $t, id, L$ ,  $\sum_{L' \in \mathcal{L}} p_\theta[id, t-1, L', L] = \sum_{L' \in \mathcal{L}} p_\theta[id, t, L, L']$ .

The above theorem provides us the ammunition needed to associate a new set of linear constraints with a pgo-theory  $\mathcal{G}$ . Our variables for this LP will correspond to each path probability:  $v_{id, t, L, L'}$ .

**Definition 9 (PLP).** For pgo theory  $\mathcal{G}$ ,  $\mathbf{PLP}(\mathcal{G})$  is the associated set of partial path based linear equations. Without loss of generality we assume the maximum time point  $T$  to be larger than any time point mentioned in  $\mathcal{G}$ .

1. Let  $\mathbf{PLP}(\cdot)$  be the constraints obtained by replacing  $p_\theta[id, t, L, L']$  with  $v_{id, t, L, L'}$  in items (1)–(3) of Theorem 1.

2. For pgo-atom  $a = (id, t, L, p)$ , let  $\mathbf{PLP}(a)$  be  $\sum_{L' \in \mathcal{L}} v_{id, t, L, L'} = p$ .

3. For in-atom  $a = in(id, r, t, p)$ , let  $\mathbf{PLP}(a)$  be  $\sum_{L \in r} \sum_{L' \in \mathcal{L}} v_{id, t, L, L'} \geq p$ .

For pgo-theory  $\mathcal{G}$ ,  $\mathbf{PLP}(\mathcal{G}) = \mathbf{PLP}(\cdot) \cup \bigcup_{a \in \mathcal{G}} \mathbf{PLP}(a)$ .

The following important theorem shows that solvability of  $\mathbf{PLP}(\mathcal{G})$  determines consistency of  $\mathcal{G}$  and that entailment of in queries can be determined by solving a linear program with  $\mathbf{PLP}(\mathcal{G})$  as the set of constraints and an objective function based on the query.

**Proposition 2.** For pgo theory  $\mathcal{G}$

- 1.  $\mathbf{PLP}(\mathcal{G})$  has a solution iff  $\mathcal{G}$  is consistent.
- 2. In query  $in(id, r, t, p)$  is a logical consequence of  $\mathcal{G}$  iff minimizing  $\sum_{L \in r} \sum_{L' \in \mathcal{L}} v_{id, t, L, L'}$  subject to  $\mathbf{PLP}(\mathcal{G})$  is greater than or equal to  $p$ <sup>1</sup>.
- 3. Checking consistency of a pgo-theory and checking entailment of ground in queries are solvable in polynomial time.

### 5.1 PLP's Size

$\mathbf{PLP}(\mathcal{G})$  is significantly smaller than that of  $\mathbf{LP}(\mathcal{G})$  because it only contains  $|ID| \cdot |T| \cdot |\mathcal{L}|^2$  variables and

$$|T| \cdot |ID| + |T| \cdot |\mathcal{L}|^2 + |T| \cdot |ID| \cdot |\mathcal{L}| + |\mathcal{G}|$$

equations. The size of this linear program is therefore bounded by  $|ID|^2 \cdot |T|^2 \cdot |\mathcal{L}|^4 \times |\mathcal{G}|$ . In contrast,  $\mathbf{LP}(\mathcal{G})$  had a size of  $|\mathcal{L}|^{|T| \cdot |ID|} \times |\mathcal{G}|$ .

Furthermore, it turns out that there are alternate ways of expressing  $\mathbf{PLP}(\mathcal{G})$  which result in more easily solvable linear programs.

### 5.2 Variable Pruning

The first simplification we can make to  $\mathbf{PLP}(\mathcal{G})$  or any set of linear equations comes when we know  $v_{id, t, L, L'}$  to be zero. In such cases, we can safely eliminate  $v_{id, t, L, L'}$  from  $\mathbf{PLP}(\mathcal{G})$ .

### 5.3 An Alternative Linear Program: ALP

Many pgo-theories only mention some time points from the set of all time points  $T$ . This can be leveraged to create a small linear program which does not reference the unmentioned time points.

**Definition 10** (Alternate Linear Program). Let  $\mathcal{G}$  be a set of pgo atoms and  $id$  be a vehicle. Let  $t_1 < t_2 \dots < t_n$  be all the time points such that for every  $t_i$  there is a pgo atom  $pgo(id, l, t_i, p) \in \mathcal{G}$ . Without loss of generality, we can assume  $t_n$  is not the maximum time point.  $\mathbf{ALP}(\mathcal{G}, id)$  is the following alternate set of linear equations:

- 1. For each  $a = pgo(id, l, t_i, p) \in \mathcal{G}$  such that  $t_1 \leq t_i < t_n$ ,  $\mathbf{PLP}(a) \in \mathbf{ALP}(\mathcal{G}, id)$ .

<sup>1</sup>If  $t$  is the maximum time point then we need to minimize  $\sum_{L \in r} \sum_{L' \in \mathcal{L}} v_{id, t-1, L', L}$ .

2. For each  $a = \text{pgo}(id, l, t_n, p) \in \mathcal{G}$ ,  
 $\sum_{L' \in \mathcal{L}} v_{id, t_{n-1}, L', l} = p \in \mathbf{ALP}(\mathcal{G}, id)$ .
3. For each  $t_1 \leq t_i \leq t_n$ ,  
 $\sum_{L \in \mathcal{L}} \sum_{L' \in \mathcal{L}} v_{id, t_i, L, L'} = 1 \in \mathbf{ALP}(\mathcal{G}, id)$
4. For each  $t_1 \leq t_i < t_{i+1} \leq t_n$ , for each  $L, L'$  such that  
 $\neg \text{reachable}(id, t_{i+1} - t_i, L, L')$  we have  
 $v_{id, t_i, L, L'} = 0 \in \mathbf{ALP}(\mathcal{G}, id)$ .
5. For each  $t_1 < t_i < t_n$ , for each  $L$  we have  
 $\sum_{L' \in \mathcal{L}} v_{id, t_{i-1}, L', L} - \sum_{L' \in \mathcal{L}} v_{id, t_i, L, L'} = 0 \in \mathbf{ALP}(\mathcal{G}, id)$

**Theorem 2.**  $\text{PLP}(\mathcal{G})$  is solvable iff  $\mathbf{ALP}(\mathcal{G}, id)$  is solvable for every  $id$ .

This theorem provides us with added efficiency in two ways. First, it prunes a fair number of variables. Second, it divides the linear program into  $|ID|$  linear programs, one for each vehicle. When the entire linear program is considered, the running time is  $O(r^3)$ , where  $r$  is the number of variables. But, if we consider each vehicle individually, the running time is proportional to  $|ID| \cdot O((r/|ID|)^3)$ , giving a speedup of  $O(|ID|^2)$ . We can further exploit this trick of dividing the linear program into smaller sub-problems by considering complete points in a pgo theory.

**Theorem 3.** For pgo theory  $\mathcal{G}$  and vehicle  $id$  complete for  $id$  at time  $t$ , let  $n$  be the max time point referenced by the theory, let  $\mathcal{G}_{0-t}^{id}$  be  $\{\mathcal{G}^{id, t'} \mid 0 \leq t' < t\}$  and let  $\mathcal{G}_{t-T}^{id}$  be  $\{\mathcal{G}^{id, t'} \mid t \leq t' < T\}$ . Then (i)  $\mathbf{ALP}(\mathcal{G}, id)$  is solvable iff  $\mathbf{ALP}(\mathcal{G}_{0-t}^{id}, id)$  and  $\mathbf{ALP}(\mathcal{G}_{t-T}^{id}, id)$  are solvable. (ii) The solution to  $\mathbf{ALP}(\mathcal{G}_{0-t}^{id}, id)$  and  $\mathbf{ALP}(\mathcal{G}_{t-T}^{id}, id)$  gives a solution for  $\mathbf{ALP}(\mathcal{G}, id)$ .

This theorem is particularly useful for in queries. Since only one time  $t$  is ever referenced by a in query, we can sandwich  $t$  between the two nearest complete time,  $t_1$  and  $t_2$  s.t.  $t_1 \leq t \leq t_2$ . Then, since we assume in queries to be asked only of consistent pgo theories, we are only required to solve for the particular  $\mathcal{G}_{t_1-t_2}^{id}$  which contains the query. In the best case, the in query will fall on a complete time/id pair, meaning that to solve the query we must only use the linear constraints in  $\mathbf{ALP}(\mathcal{G}^{id, t}, id)$  (where  $t$  is the in query's time point and  $id$  is the query vehicle).

## 5.4 ALP Size

Now, instead of solving PLP, a massive linear program, we are solving many smaller  $\mathbf{ALP}(\mathcal{G}_{t_1-t_2}^{id}, id)$ .

For vehicle  $id$  in theory  $\mathcal{G}$  with  $n$  mentioned time points of which every  $c^{\text{th}}$  one is complete, and each two consecutive time points,  $t_1$  and  $t_2$ , we must solve  $\mathbf{ALP}(\mathcal{G}_{t_1-t_2}^{id}, id)$ . This will happen in  $O(n^3)$  where  $n$  is the number of variables. The number of variables is bounded by  $(c+1) \cdot |\mathcal{L}|^2$ . Thus computing consistency of  $\mathcal{G}$  using  $\mathbf{ALP}$  will happen in time  $|ID| \times n/c \times O(((c+1) \cdot |\mathcal{L}|^2)^3)$ . There is an interesting phenomenon happening here: the running time of our algorithms may decrease as the size of the theory grows – so long as more complete time points are added.

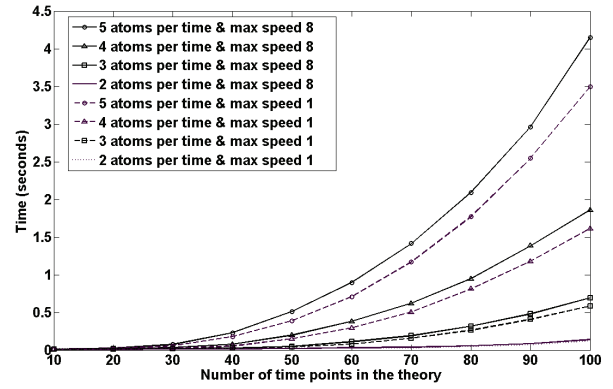


Figure 1: Time to check consistency of complete theories with 2 to 5 pgo atoms per time point when maximum speed is 1 and 8.

## 6 Experimental Section

We have implemented a prototype pgo system in MatLab on a Pentium 4 (3.80GHz) processor running under Windows XP and with 2GB of memory. Our system implements all algorithms described in this paper for both complete and incomplete theories.

We ran several experiments to test the performance of these algorithms and identify the important factors that affect the performance other than the obvious ones such as number of atoms and time points referenced. We performed our experiments on theories that refer to a single vehicle using  $\mathbf{ALP}$  type linear programs.

The maximum number of pgo atoms per time point in a complete theory plays an important role in checking consistency. This number gives a maximum number of path probabilities for each time point. Another important factor we considered was the maximum speed of the vehicle because this affects the maximum number of reachable locations and hence the total number of path probability variables in our final linear program. To test the effect of these speed and atom density, we created random theories in a  $50 \times 50$  grid. We varied the maximum number of atoms per time point from 2 to 5, with maximum speeds of 1 and 8. The number of distinct time points in the theory varied from 10 to 100. We derived the speed values as follows: suppose we use the  $50 \times 50$  grid to represent the USA. Then speed of 1 will coincide with that of a car (70 mph) while a speed of 8 will coincide with that of a plane (500 mph).

**Consistency check time for complete theories:** Figure 1 shows the time taken for consistency checking for 8 kinds of complete theories. The data points represent the average over 50 randomly generated theories. As seen in the figure the effect of increasing number of pgo atoms per time point has a greater impact on performance than increasing maximum speed. The reader can see that it only takes a few seconds to reason about 100 time points.

**Consistency check time for incomplete theories:** For incomplete theories, the size of the grid has a great impact on the time required to check consistency. We say a time point

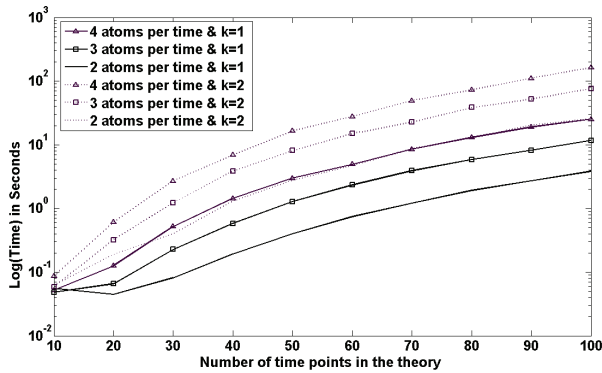


Figure 2: Time to check consistency of incomplete theories with 2, 3, and 4 pgo atoms per time point, a maximum speed of 1 or 8, and *Incmax* varying from 1 to 2 (displayed as *k* in the graph’s key).

*t* is *incomplete* iff the sum of the probability field in all pgo atoms ends up being less than 1. We investigated how the structure of the theory affects the consistency checking algorithm. Let *Incmax* be the maximal number of incomplete time points followed by a complete time point in a theory. For this experiment we created random theories in a 10 by 10 grid with maximum speed 1, maximum number of pgo atoms per complete time point ranging from 2 to 4, *Incmax* = 1 or *Incmax* = 2 and the number of referenced time points ranging from 10 to 100. Furthermore every complete time point is followed by *Incmax* incomplete time points in the theory. So when *Incmax* = 1 and the total number of time points in the theory is 50, there are 25 incomplete time points interleaved with 25 complete points. Figure 2 shows the time taken to check consistency for 6 kinds of randomly generated theories. The data points represent the average of 20 runs — note that the *y*-axis uses a logarithmic scale. As seen in the figure, the effect of increasing the number of pgo atoms per time point has a similar effect on the performance. However, increasing *Incmax* affects the running time dramatically.

**In-queries.** Figure 3 shows the time required to answer “in” queries as we vary the temporal density of a complete theory. Temporal density of a theory is the ratio of time points referenced in the theory and the total number of time points. For these experiments we set the grid size to 25 by 25 and maximum time points to 500 and number of pgo atoms per time point to 3. For example when the theory has a temporal density of 1, it has a total of 1500 time points. The data points in the graph are an average of 100 runs. The reader can easily see that the time taken drops exponentially with an increase in the density. Since a rise in density corresponds to an increase in theory size, these results are particularly interesting yet consistent with Theorem 3. It shows our algorithms’ running time *decreases* as the number of referenced time points increases. This is sensible: when one is working with probabilistic data, one should sometimes find it easier to answer queries as the amount of data increases, because fewer possible satisfying interpretations for the data need be considered.

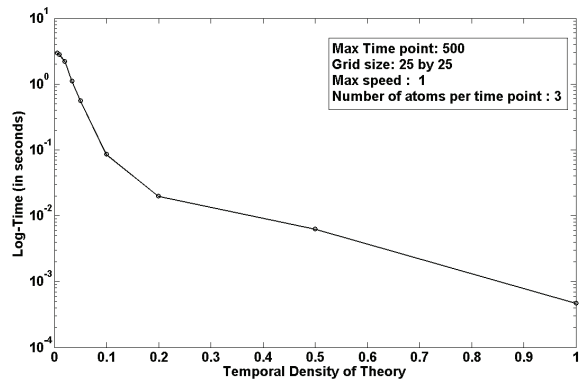


Figure 3: Time to answer in-queries w.r.t. complete theories of varying temporal density.

## 7 Related Work

There are several *spatio-temporal logics* [Gabelaia *et al.*, 2003; Merz *et al.*, 2003; Wolter and Zakharyashev, 2000; Cohn *et al.*, ] in the literature. These logics extend temporal logics to handle space. Most of them involve logical languages similar to LTL. There is also much work on qualitative spatio-temporal theories (for a survey see [Anthony G. Cohn, 2001]). The closest work to ours is that of [Muller, 1998a; 1998b] which describes a formal theory for reasoning about motion in a qualitative frame work. The expressive power of the theory allows for the definition of complex motion classes. The focus of these works is qualitative - in contrast, we deal with uncertainty about where vehicles will be in the future. Our methods are rooted in a mix of probability, geometry and logic rather than just logic alone.

Other related work includes [Shanahan, 1995] which 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 rooted in a mix of geometry and logic rather than logic alone.

## 8 Conclusions

There are numerous applications where we wish to reason about moving objects. In some cases, (i) we know the intended destinations of the moving objects while in others (ii) we do not know this for sure. In this paper, we have developed the concept of a probabilistic go theory (pgo theory), which is capable of handling both cases. This extends the notion of a go-theory proposed by [Yaman *et al.*, 2004], which only handles case (i).

As future work, we intend to examine aggregation techniques which would allow the creation of minimally sized linear programs solving pgo problems. We further intend to include queries which reference periods of time. Also, work must be done to expand this system to handle multi-object

queries where the vehicles' locations are not independent of one another.

We have presented a syntax and declarative semantics for pgo theories, efficient algorithms to check the consistency of pgo theories, and efficient algorithms to implement "in" queries. Our implementation shows that these algorithms work effectively in practice.

## 9 Acknowledgments

This work was partly supported by AFOSR grants FA95500610405 and FA95500510298, ARO grant DAAD190310202 and by the Joint Institute for Knowledge Discovery.

## References

- [Anthony G. Cohn, 2001] Shyamanta M. Hazarika Anthony G. Cohn. Qualitative spatial representation and reasoning: An overview. *Fundam. Inform.*, 46(1-2):1–29, 2001.
- [Chen and Chien., 2001] M. Chen and S. Chien. Dynamic freeway travel prediction time using probe vehicle data: Link-based vs. path-based. In *Conf. of the Transportation Research Board*, Washington DC., 2001.
- [Cohn *et al.*, ] 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 C. Freksa, C. Habel, and K.F Wender, editors, *Spatial Cognition III*, Lecture Notes in Computer Science. Springer-Verlag. to appear.
- [Gabelaia *et al.*, 2003] David Gabelaia, Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyashev. On the computational complexity of spatio-temporal logics. In I. Russell and S. Haller, editors, *Proceedings of the 16th AAAI International FLAIRS Conference*, pages 460–464. AAAI Press, 2003.
- [Kato *et al.*, 2004] J. Kato, T. Watanabe T., S. Joga, Y. Liu, and H. Hase. An hmm/mrf based stochastic framework for robust vehicle tracking. *IEEE Transactions on Transportation Systems*, 5(3):142–154, Sep 2004.
- [Merz *et al.*, 2003] Stephan Merz, Júlia Zappe, and Martin Wirsing. A spatio-temporal logic for the specification and refinement of mobile systems. In Mauro Pezzè, editor, *Fundamental Approaches to Software Engineering (FASE 2003)*, volume 2621 of *Lecture Notes in Computer Science*, pages 87–101, Warsaw, Poland, April 2003. Springer-Verlag.
- [Mittu and Ross, 2003] R. Mittu and R. Ross. Building upon the coalitions agent experiment (coax) - integration of multimedia information in gccs-m using impact. In *Proceedings 9th International Workshop on Multimedia Information Systems*, pages 35–44, 2003.
- [Muller, 1998a] Philippe Muller. A qualitative theory of motion based on spatio-temporal primitives. In Anthony G. Cohn, Lenhart Schubert, and Stuart C. Shapiro, editors, *KR'98: Principles of Knowledge Representation and Reasoning*, pages 131–141, San Francisco, California, 1998. Morgan Kaufmann.
- [Muller, 1998b] Philippe Muller. Space-time as a primitive for space and motion. In *FOIS'98*, pages 63–76, Amsterdam, 1998.
- [Rajagopalan and Kuipers, 1994] R. Rajagopalan and B. Kuipers. Qualitative spatial reasoning about objects in motion: Application to physics problem solving. In *IEEE Conf. on AI for Applications*, pages 238–245, San Antonio, 1994.
- [Shanahan, 1995] Murray Shanahan. Default reasoning about spatial occupancy. *Artificial Intelligence*, 74(1):147–163, 1995.
- [Tsang *et al.*, 1999] S.H. Tsang, E.G. Hoare, P.S. Hall, and N.J. Clarke. Automotive radar image processing to predict vehicle trajectory. In *Intl. Conf. on Image Processing*, volume 3, pages 867–870, Kobe Japan, 1999.
- [Wolter and Zakharyashev, 2000] Frank Wolter and Michael Zakharyashev. Spatio-temporal representation and reasoning based on RCC-8. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *KR2000: Principles of Knowledge Representation and Reasoning*, pages 3–14, San Francisco, 2000. Morgan Kaufmann.
- [Yaman *et al.*, 2004] Fusun Yaman, Dana Nau, and V S Subrahmanian. A logic of motion. In *Proceedings of KR2004*, pages 85–94, 2004.
- [Yaman *et al.*, 2005] Fusun Yaman, Dana Nau, and V S Subrahmanian. Going far, logically. In *Proceedings of IJCAI2005*, 2005.