ABSTRACT

Title of dissertation: DECLARATIVE REASONING ABOUT MOVING OBJECTS
Fusun Yaman Sirin, Doctor of Philosophy, 2006

Dissertation directed by: Dana S. Nau
Department of Computer Science

There are numerous applications where there is a critical need to reason about moving object plans under uncertainty. Previous work on spatio-temporal logics is limited to qualitative approaches and the work on spatio-temporal databases focus on the observations ignoring the intended movements of the objects. This thesis presents a *Logic of Motion (LOM)*, a novel theory and algorithms that combine logic, constraint satisfaction and geometric reasoning. LOM provides a *declarative syntax and model theory* and formalizes how to reason about planned movements of objects, when there is uncertainty. LOM is the first quantitative logical treatment of moving objects that can account for the fact that we are not always sure when an object will leave or arrive a given location, and what its velocity will be.

The thesis includes the following contributions:

- LOM, the first quantitative logic to reason about flexible plans for moving objects.
- An analysis of the computational complexity of reasoning with flexible plans for moving objects. This analysis includes an important theoretical result showing that

complexity of consistency checking for LOM theories is at least NP-hard. I also provide algorithms to check consistency of a fraction of LOM theories called go-theories.

- A class of motion theories, called Simple Go-Theories that are tractable.
- Efficient algorithms to answer ground and non-ground queries in LOM concerning the possible location of the object and its proximity to other objects.
- A study of default reasoning for motion-theories. It presents a *motion closed world assumption* for LOM that restrict the reasoning within a class of preferred models of the theory. Motion closed world assumption allows us to make more intelligent and customized inferences.
- An investigation of deconfliction of motion-theories with respect to some integrity constraints. A deconfliction of a theory is a modification to the theory such that any model of the modified theory will entail the integrity constraints. I present an algorithm for efficiently computing a deconfliction of a theory.
- Extensive empirical evaluation to demonstrate the efficiency of consistency checking, query answering and deconfliction algorithms.

DECLARATIVE REASONING ABOUT MOVING OBJECTS

by

Fusun Yaman Sirin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Commmittee:

Professor Dana S. Nau, Chair
Professor Michael Fu
Professor Atif Memon
Professor Hector Munoz-Avila
Professor V.S. Subrahmanian

DEDICATION

To my family...

ACKNOWLEDGMENTS

I am grateful to my advisor and my role model Professor Dana S. Nau for his guidance in every aspect of my life. He always believed in my potential and if it was not for his trust I would not be able to complete this thesis. Also I sincerely thank to Professor V.S. Subrahmanian who suggested my dissertation topic and supported my research both intellectually and financially. I am especially thankful to him for financing several of my conference attendances that helped me expose my work to a wider audience.

I can not thank enough to my beloved husband Evren Sirin for making my life worthy of living throughout my PhD. Without his loving care and patience I would not survive through this painful process.

I am also grateful to my parents Havva Yaman and Orhan Yaman who made my education their top priority and lead me to where I am right now.

I also need to acknowledge the support of my friends Burcu Karagol-Ayan, Fazil Ayan, Mustafa Tikir and Okan Kolak in making the life in College Park more fun and bearable. Special thanks to the Ayan family and their little son Alperen for proving me that having a family and a PhD can coexist.

Finally I thank my little daughter Gizem for being a healthy and happy baby despite all the stress she endured in my tummy while I was writing this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Reasoning about plans for moving objects is an important problem for many applications, such as air traffic management, transportation applications and military mission planning.

Here are two examples to illustrate the kinds of reasoning tasks that are required and the representational capabilities that are needed in the plans:

Air traffic management Airlines come up with flight plans for their planes; they decide on the the destination, the origin of the flights and the routes of these flights are determined by one of the FAA established air corridors. The flight plan contains the *tentative* schedule of the plane, but the tentative schedule can not always be met. For example, it is common for planes to take off later than their scheduled time due to the air traffic or severe weather conditions. However, a late takeoff does not automatically translate into a late landing, as the planes can sometimes fly fast enough to make up for the time lost on the ground. Thus the flight plans actually are more flexible than they seem on our tickets. They inherently contain some temporal flexibility with respect to departure and arrival times as well as flexibility on the the speed of the planes. For such flight plans one may want to answer questions such as:

- Is there a way to achieve all the flight plans with temporal and speed uncertainties?

- Given new information, e.g. a storm, which of the planes need to change their flight plans in order to avoid the storm?
- Which flights will be closer than a certain distance during the severe weather?
- How should the flight plans be modified to prevent the planes from entering a region during a storm?

Amphibious Mission Planning An amphibious operation [1] is a military operation launched from the sea by naval and landing forces embarked in ships or craft involving a landing on a hostile or potentially hostile shore. The complexity of amphibious operations and the vulnerability of forces engaged in these operations require an exceptional degree of unity of effort and operational coherence both in the planning and execution levels. Amphibious missions are planned by various number of officers. They plan ship-to-shore movement routes and schedules, which can involve a large number of (air and surface) vehicles and are subject to many constraints (e.g., minimum inter-ship distances, vehicle landing orderings). As a result of multiple plan editing, the merged plan often contains constraint violations, e.g. ships passing too close to each other. These violations are detected and fixed manually by looking at a visualization of the execution simulation for the plan. During the execution of the plan, the conditions may change due to enemy activity and/or weather, hence the plans need to be flexible enough to accommodate those changes. The landing times as well as the speeds of the vehicles are constrained by the mission parameters and the physical properties of the vehicles. The plans must be able to represent these constraints. Also, given new intelligence and the plans in execution, it is often important to examine the possible locations of the vehicles and their proximity to enemy. The following are examples of frequently encountered tasks in this application:

- Check if it is possible for all ships to get their destination without violating the mission constraints.
- Given an estimated position for an enemy ship, find all the ships that might be in danger, i.e. within the firing range of enemy.
- Modify the plans so that all ships will stay out of a critical no-go region, e.g., possibly a region containing an enemy submarine.
- During a critical time period , e.g. an air assault, check if all small ships will be close enough to a destroyer that can offer them protection?
- Determine whether ships will be far away from a certain enemy vehicle whose path has just been discovered.

As both of these examples clearly demonstrate, a general framework for reasoning about plans for moving objects will need to

- Represent moving object plans such that temporal and speed uncertainties can be expressed and modeled explicitly.
- Check the executability of these plans, i.e. is there a schedule that will accommodate all movement constraints in the plans?
- Answer several kinds of queries related to the location of th object and its proximity to other objects

- Deconflict these plans with respect to some constraints.

Of the previous work on spatio-temporal reasoning that has been done by logic and database researchers, none is adequate to address the challenges raised by reasoning about plans for moving objects. On one hand, the existing work done by logicians on spatio-temporal logics are all qualitative [4, 54, 87, 18], i.e. they concentrate on symbolic reasoning instead of numeric and geometric reasoning. It is impossible to represent the constraints of the movements such as the temporal and speed intervals which is critical for many real world applications. Moreover, existing logics fail to provide any efficient algorithms to perform consistency checking or answer any queries. On the other hand, spatio-temporal database researchers focus on designing data structures to enable efficient storage and querying of previously recorded observations. The main characteristic of the existing spatio-temporal databases is that they do not acknowledge the intent of the objects, i.e. that the objects are moving according to a plan. Instead they deal with observed locations and times, and make short-term future estimates on the locations of objects based on observations. Majority of the work addresses uncertainty about the route of the object, but ignores temporal or speed uncertainty.

1.2 Proposed Solution

This thesis presents a novel theory that combines logic, constraint satisfaction and geometric reasoning. It presents a quantitative *Logic of Motion* (LOM) with a formal syntax and model theoretic semantics. LOM models the temporal and speed uncertainty of the object movements explicitly. The thesis investigates the computational complexity of reasoning with such uncertainties and aims to identify tractable classes of the problem. As a first step toward developing a more expressive language it identifies three kinds of queries and provides very efficient algorithms that uses complex geometric reasoning: **(i)** in queries ask if a vehicle is guaranteed to be inside a given region in a given time interval. **(ii)** near queries ask if two objects are guaranteed to be within a given distance of each other throughout a given interval. **(iii)** far queries ask if two objects are guaranteed to be sufficiently apart from each other throughout a given interval. Note that these three queries cover most of the important queries in the motivating examples.

Finally the thesis provides a general theory and algorithms to deconflict the movement plans with respect to a set of integrity constraints. The idea is to make sure that every execution of the given plans would achieve the integrity constraints. When plans do not guarantee the satisfaction of integrity constraints, we must check to see if there is a way of strengthening the plans (i.e. make the plans less flexible than the original one) so that the integrity constraints are always satisfied.

The declarative semantics of LOM makes it suitable for many applications and allows it to be open to various extensions such as addition of new queries and customization for different applications. Moreover the semantics of LOM lets us couple it with already existing logics, to create more expressive languages. For example if we couple LOM with a description logic and spatial logic we can express queries of the form: "Find me all the *commercial planes* flying over *France*" where the concept of a commercial plane is given in the description logic and coordinates of France is described in a spatial logic.

1.3 Contributions

The contributions of this thesis are as follows:

- It provides a general framework for reasoning about plans for moving objects under uncertainty which is critical in many applications such as air traffic management and military mission planning. Using the declarative semantics it is also possible to customize this framework for different applications and couple it with other existing logics to create a more powerful theory.
- It describes the first quantitative logic to represent movements of objects while explicitly modeling temporal and speed uncertainty. Thus it bridges the gap between geometric reasoning and spatio-temporal logics.
- It presents theoretical results regarding the complexity of the problem and identifies a tractable class of problems.
- It provides very efficient algorithms to reason about the flexible movement plans and an implementation that shows the effectiveness of the algorithms.
- It presents a default reasoning mechanism for incomplete theories thus allowing more intelligent inferences.
- It describes a theory and algorithms to deconflict plans for moving objects w.r.t. certain integrity constraints.

1.4 Thesis Outline

This thesis is organized as follows:

- Chapter 2 presents a formal syntax for *motion* theories called LOM and a fragment of motion theories called *go*-theories. Also in Chapter 2 a formal model theory for motion theories is presented

- Chapter 3 provides a sound and complete algorithm to check consistency of go-theories together with a result that checking consistency of *go*-theories is NP-complete.

- Chapter 4 identifies a subclass of *go*-theories called *simple go*-theories for which consistency checking and query answering takes polynomial time. It also presents sound and complete algorithms for checking consistency of simple *go*-theories.

- Chapter 5 defines the notions of temporal and positional certainty intervals which are the building block for query answering algorithms.

- Chapter 6 presents sound and complete algorithms to verify entailment of ground atoms by go-theories.

- Chapter 7 shows how to compute answer substitutions for *certain types* of non-ground queries.

- Chapter 8 defines a Motion Closed World Assumption for a default reasoning with theories that do not cover all time points. It provides algorithms to reason with the closed world assumption

- Chapter 9 augments go-theories with integrity constraints and defines deconfliction of theories with respect to these integrity constraints. It also presents an algorithm to find deconflictions.
- Chapter 10 describes a prototype implementation and presents numerous experiments that show the that go-theories are practical to use.
- Chapter 11 discusses related work
- Finally Chapter 12 concludes with the summary, impact of the thesis and discussions about the future work.

Chapter 2

Syntax and Semantics of LOM

2.1 Syntax

We assume the existence of three sets of constant symbols: $\mathbf{R}$ is the set of all real numbers, $\mathbf{OID}$ is the set of all object ids and $\mathbf{P}$ is the set of all points in a three dimensional cartesian space. We assume the existence of three disjoint sets $V_{\mathbf{R}}, V_{\mathbf{OID}}, V_{\mathbf{P}}$ of variables ranging over $\mathbf{R}$, $\mathbf{OID}$ and $\mathbf{P}$, respectively. We also assume the existence of four special predicate symbols $\mathsf{go}, \mathsf{near}, \mathsf{far}, \mathsf{in}$, of arities $9, 5, 5, 5$ respectively. As usual a *real* (resp. object) term t (resp. o) is any member of $\mathbf{R} \cup V_{\mathbf{R}}$ (resp. $\mathbf{OID} \cup V_{\mathbf{OID}}$). A *point* term p is any member of $\mathbf{P} \cup V_{\mathbf{P}}$ and p^x, p^y, p^z represents the values of the x, y and z components of p respectively.

LOM Atoms. Let P_1, P_2 be point terms and t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+, d, t_1, t_2 be real terms; and let o, o_1, and o_2 be object terms. Then the following are *LOM atoms*:

$$\mathsf{near}(o_1, o_2, d, t_1, t_2);$$

$$\mathsf{far}(o_1, o_2, d, t_1, t_2);$$

$$\mathsf{in}(o, P_1, P_2, \ell, h);$$

$$\mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+).$$

Ground terms and ground atoms are defined in the usual way.

Intuitively, $\mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ is true if object o leaves location P_1

at some time point in the interval $[t_1^-, t_1^+]$, goes ***along the straight line connecting*** P_1 ***and*** P_2 at a speed between v^- and v^+, and arrives at location P_2 at some time point in the interval $[t_2^-, t_2^+]$. Note that the object is not allowed to stray from the straight line connecting P_1 and P_2 - moreover, our semantics will prevent the object from making strange movements (such as travelling part way from P_1 to P_2, reversing direction for a bit, and then re-reversing direction and heading towards P_2).

Intuitively, $\mathsf{in}(o, P_1, P_2, t_1, t_2)$ is true if o is guaranteed to be inside the cuboid volume whose lower-front left corner is P_1 and whose upper-back right corner is P_2 (with edges parallel to x, y, z axes) at some time point between t_1, t_2.

$\mathsf{near}(o_1, o_2, d, t_1, t_2)$ is true if objects o_1 and o_2 are within distance d of each other at *all* times t, $t_1 \leq t \leq t_2$.

Finally, $\mathsf{far}(o_1, o_2, d, t_1, t_2)$ is true if objects o_1 and o_2 are not within distance d of each other at *all* times t, $t_1 \leq t \leq t_2$

Note that unlike most existing studies of motion (whether in the AI or database), this framework allows us to express uncertainty about when an object leaves a given location, when it arrives at its destination, and what its velocity is. This is consistent with the real world where the velocity of an object may vary (e.g. with traffic in the case of road vehicles, with wind speed in the case of aerial vehicles, with oceanographic currents in the case of marine vehicles, etc.). This in turn has an impact on exactly when a vehicle will reach its destination - that too is uncertain. There is no existing treatments of this uncertainty in reasoning about moving objects.

Definition 1 (LOM-formula) *LOM formulas are inductively defined as follows:*

- *Every* LOM *atom is a* LOM *formula;*
- *if* F_1, F_2 *are* LOM *formulas, then so are* $F_1 \wedge F_2$, $F_1 \vee F_2$ *and* $\neg F_1$.

Definition 2 (Motion/Go Theory) *A* motion theory *is a finite set of* LOM *formulas. A* go-*theory is a finite set of* ground go-*atoms.*

Notation. If

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

then we will let

$$obj(g) = o, \quad v^-(g) = v^-, \quad v^+(g) = v^+,$$

$$loc_1(g) = P_1, \quad t_1^-(g) = t_1^-, \quad t_1^+(g) = t_1^+,$$

$$loc_2(g) = P_2, \quad t_2^-(g) = t_2^-, \quad t_2^+(g) = t_2^+.$$

Also $LS(g)$ is the line segment $[P_1, P_2]$. We use $\mathsf{dist}(P_1, P_2)$ to denote the Euclidean distance between P_1 and P_2, i.e. $\mathsf{dist}(P_1, P_2) = \sqrt{(P_2^x - P_1^x)^2 + (P_2^y - P_1^y)^2 + (P_2^z - P_1^z)^2}$.

Definition 3 *Let* $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *be a go atom. The* ***direction*** *of* g *is a unit vector* $\vec{v}(g) = [x, y, z]$ *such that*

- $x = (P_2^x - P_1^x)/d$
- $y = (P_2^y - P_1^y)/d$
- $z = (P_2^z - P_1^z)/d$

where $d = \mathsf{dist}(P_1, P_2)$

2.2 Semantics: Model Theory

In this section, we define a formal model theoretic semantics for LOM. A *LOM-interpretation* I is a continuous[1] function from $\mathbf{OID} \times \mathbf{R}$ to $\mathbf{P}$. Intuitively, $I(o,t)$ is the location of object o at time t.

Definition 4 *Let* $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *be an atom and* I *be an interpretation.* I ***satisfies*** g *over a time interval* $T = [t_1, t_2]$ *iff:*

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

The above definition intuitively says that $I \models g$ over a time interval $T = [t_1, t_2]$ iff o starts moving at t_1, stops moving at t_2 and during this interval, 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.

Definition 5 (Satisfaction) *Suppose* F *is a formula and* I *is a LOM-interpretation. We say that* I satisfies F*, denoted* $I \models F$*, iff:*

[1]Continuity is w.r.t. classical real fields [29] rather than, say, the discrete notion of continuity used in logic programming [50].

1. *If* $F = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$*, then* $I \models F$ *iff there exists an interval* T *such that* $I \models F$ *over the interval* T

2. *If* $F = \mathsf{near}(o_1, o_2, d, t_1, t_2)$*, then* $I \models F$ *iff* $\mathsf{dist}(I(o_1, t), I(o_2, t)) \leq d$ *for every* $t_1 \leq t \leq t_2$*.*

3. *If* $F = \mathsf{in}(o, P_1, P_2, t_1, t_2)$*, then* $I \models F$ *iff there are reals* $t \in [t_1, t_2]$*,* $x \in [{P_1}^x, {P_2}^x]$*,* $y \in [{P_1}^y, {P_2}^y]$ *and* $z \in [{P_1}^z, {P_2}^z]$ *such that* $I(o, t) = (x, y, z)$*.*

4. $I \models F \wedge G$ *iff* $I \models F$ *and* $I \models G$*.*

5. $I \models F \vee G$ *iff* $I \models F$ *or* $I \models G$*.*

6. $I \models \neg F$ *iff* I *does not satisfy* F*.*

I satisfies *a motion theory* MT *iff* I *satisfies every* $F \in \mathsf{MT}$.

MT is *consistent* iff there is a LOM interpretation I such that $I \models \mathsf{MT}$. F is a *logical consequence* of MT, denoted $\mathsf{MT} \models F$, iff every LOM interpretation I that satisfies MT also satisfies F.

Suppose $G = \{g_1, g_2\}$, where

$$g_1 = \mathsf{go}(o, P_{11}, P_{12}, t_{11}^-, t_{11}^+, t_{12}^-, t_{12}^+, v_1^-, v_1^+);$$

$$g_2 = \mathsf{go}(o, P_{21}, P_{22}, t_{21}^-, t_{21}^+, t_{22}^-, t_{22}^+, v_2^-, v_2^+).$$

Suppose the points $P_{11}, P_{12}, P_{21}, P_{22}$ are all distinct and $t_{21}^- > t_{12}^+$. Then the above definition allows I to have the object o travel at an arbitrarily high speed during the open interval (t_{12}^+, t_{21}^-).

Chapter 3

Consistency Checking

3.1 Problem

Checking consistency of a go-theory is complicated by the fact that a single go atom can be inconsistent. For instance, consider $\mathsf{go}(o, P_1, P_2, 44, 48, 50, 52, 4, 5)$, where $P_1 = (0, 0, 0)$ and $P_2 = (0, 60, 0)$. This go atom is inconsistent because there is no way to get from P_1 to P_2 within the prescribed speed limits of 4-5 miles per hour within the prescribed time frame (leave sometime in the $[44, 48]$ interval and arrive at some time in the $[50, 52]$ interval). In addition, consider the go-atom $\mathsf{go}(o, P_1, P_2, 35, 50, 40, 55, 6, 10)$ with same P_1 and P_2 as before. The distance between the origin and destination is 60. If the object leaves at time 35, the earliest time at which it can arrive at the destination (given the max speed of 10) is at time 41. Thus the lower bound on the arrival time which is 40 in the above go-atom can be tightened to 41. Likewise, the upper bound may also be amenable to tightening. The following definition generalizes this idea to define the normalization of a go atom, such that normalization of a go atom has exactly same set of solutions but tighter bounds on arrival and departure times.

Definition 6 (Normalization) *Let* $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *and* $g^* = \mathsf{go}(o, P_1, P_2, T_1^-, T_1^+, T_2^-, T_2^+, V^-, V^+)$ *be two go atoms.* g^* *is the* ***normalization*** *of* g *iff the following hold:*

- $t_1^- \leq T_1^- \leq T_1^+ \leq t_1^+$
- $t_2^- \leq T_2^- \leq T_2^+ \leq t_2^+$
- $v^- \leq V^- \leq V^+ \leq v^+$
- *I is a model of g iff I is a model of g^**
- *If $g' = \mathsf{go}(o, P_1, P_2, T_3^-, T_3^+, T_4^-, T_4^+, V_3^-, V_3^+)$ is a go-atom that satisfies all the above conditions, then $[T_1^-, T_1^+] \subseteq [T_3^-, T_3^+]$ and $[T_2^-, T_2^+] \subseteq [T_4^-, T_4^+]$ and $[V^-, V^+] \subseteq [V_3^-, V_3^+]$.*

Clearly every go atom in a consistent *go*-theory has a normalization.

Assumption. Throughout the rest of this thesis, we will assume, without loss of generality, that all go-atoms are normalized.

Assumption. Given a *go*-theory G and an object o, let G^o denote the set of all atoms $g \in G$ such that $obj(g) = o$. It is clear that G is consistent iff G^o is consistent for all objects o. *Therefore, throughout the rest of this chapter, without loss of generality, we assume that for all $g_i, g_j \in G$, $obj(g_i) = obj(g_j)$. In other words, all atoms in G are about the same object.*

3.2 Complexity

The following theorem describes the computational complexity of checking consistency.

Theorem 1 *The problem of checking whether an input motion theory is consistent is NP-hard. The problem is NP-complete if the input theory is a* go*-theory.*

Proof The proof of NP-hardness is obtained by polynomially reducing the problem of sequencing with release times and deadlines (***SRD***) which is known to be NP-complete [34]. For this problem

- Input: A set T of tasks; for each task t in T: a positive integer length $len(t)$; a positive integer release time $r(t)$; and a positive integer deadline $d(t)$.

- Question: Is there a one-processor schedule for the tasks, T, that satisfies the release time constraints and meets all of the deadlines, i.e. a one-to-one function σ from the set of tasks to positive integers such that all of the following hold:

  – For any two distinct tasks t and w if $\sigma(t) > \sigma(w)$ then $\sigma(t) \geq \sigma(w) + len(w)$.

  – For all tasks t in T, $\sigma(t) \geq r(t)$.

  – For all tasks t in T, $\sigma(t) + len(t) \leq d(t)$?

Let $o \in$ **OID** then for any given SRD-problem with $T = \{t_1..t_k\}$, the corresponding *go*-theory $G = \{g_1...g_k\}$ can be constructed as follows:

For all $1 \leq i \leq k$

- $obj(g_i) = o$

- $loc_1(g_i) = P_{1i}$ and $loc_2(g_i) = P_{2i}$ where P_{1i} andP_{2i} are arbitrary points

- $t_1^-(g_i) = r(t_i)$ and $t_1^+(g_i) = d(t_i) - len(t_i)$

- $t_2^-(g_i) = r(t_i) + len(t_i)$ and $t_2^+(g_i) = d(t_i)$

- $v_1(g_i) = v_2(g_i) = dist(P_{1i}, P_{2i})/len(t_i)$

Clearly if G is consistent than there is a one processor schedule for the tasks in T that satisfy the properties above.

Later in this chapter, Section 3.5 presents a nondeterministic polynomial time algorithm that checks the consistency of a *go*-theory. The existence of the algorithm proves that checking consistency of a *go*-theory is NP-complete.

□

3.3 Movements

Consider the following question: *Given a set G of* go*-atoms, is it possible for G to describe the movement of a single object o on a single line segment over a single continuous time interval?* In this case, we say that the set G of atoms describes a coherent movement. Consider the case where G consists of two go-atoms g_1, g_2:

1. Suppose $t_2^+(g_1) < t_1^-(g_2)$. Then the answer is no, because g_1 must end before g_2 starts.
2. Suppose $t_1^-(g_1) \leq t_2^+(g_2)$, $t_1^-(g_2) \leq t_2^+(g_1)$ and the line segments $LS(g_1)$ and $LS(g_2)$ are either not collinear or do not intersect or have different directions. Then the answer is no because even though g_1 and g_2 overlap temporally, they define incompatible trajectories for o during the overlapping time.
3. Suppose $t_1^-(g_1) \leq t_2^+(g_2)$, $t_1^-(g_2) \leq t_2^+(g_1)$ and the line segments $LS(g_1)$ and $LS(g_2)$ are collinear and their intersection is nonempty and they have the same direction. Then the answer may be either yes or no, depending on whether the minimum and maximum speeds $v^-(g_1), v^+(g_1), v^-(g_2), v^+(g_2)$ are compatible and

if the common line segment can be visited at the same time by both g_1 and g_2. Since o's actual speed does not need to be constant, those conditions are rather complicated.

Definition 7 (Compatibility/Concurrency) *Suppose o is an object and suppose $g, g' \in G^o$. g and g' are **compatible** iff there is an interpretation I and there are time intervals $[t_1, t_2]$ and $[t'_1, t'_2]$ such that*

- *I satisfies g over $[t_1, t_2]$*
- *I satisfies g' over $[t'_1, t'_2]$*
- *$t_2 > t'_1$ and $t'_2 > t_1$*

*If I is such an interpretation, then I is said to **concurrently** satisfy both g and g'.*

Example 1 *Let $g_1 = go(o, (40, 10, 55), (40, 60, 55), 1, 5, 6, 14, 5, 10)$ and $g_2 = go(o, (40, 30, 55), (40, 90, 55), 8, 10, 13, 18, 6, 12)$. g_1 and g_2 are compatible because there is a LOM interpretation I that satisfies g_1 over time interval $[4, 13]$ and g_2 over $[8, 16]$. Hence I satisfies g_1 and g_2 concurrently.*

The following lemma presents the necessary conditions for two go atoms to be compatible.

Lemma 1 *If two go atoms g_1, g_2 are **compatible** then*

- *The directions of g_1 and g_2 are the same (i.e $\vec{v}(g_1) = \vec{v}(g_2)$) and*
- *The intersection of $LS(g_1)$ and $LS(g_2)$ is a line segment, and*
- *The atoms temporally overlap, i.e. $t_1^-(g_1) \le t_2^+(g_2)$ and $t_1^-(g_2) \le t_2^+(g_1)$*

- *The speed intervals overlap, i.e.* $v^-(g_1) \leq v^+(g_2)$ *and* $v^-(g_2) \leq v^+(g_1)$

The above conditions are called ***compatibility conditions****.*

Example 2 *Let* g_1 *and* g_2 *be the two compatible atoms defined in Example 1.* g_1 *and* g_2 *satisfy all the compatibility conditions because:*

- *The directions of* g_1 *and* g_2 *are the same:* $\vec{v}(g_1) = \vec{v}(g_2) = (0, 1, 0)$*,*
- $LS(g_1) \cap LS(g_2)$ *is the line segment* $[(40, 30, 55), (40, 60, 55)]$*,*
- *The atoms temporally overlap:* $1 \leq 18$ *and* $8 \leq 14$
- *The speed intervals overlap:* $5 \leq 12$ *and* $6 \leq 10$*.*

We are now going the generalize the notion of compatibility and concurrency to a set of go atoms using the concept of a concurrency graph.

Definition 8 (Concurrency graph and movements) *Let* $G = \{g_1, g_2 \ldots g_n\}$ *be a go theory. A graph* $\Gamma = \langle V, E \rangle$ *is a concurrency graph for* G *iff* $V = G$ *and* $E \subseteq \{(g_i, g_j) \in E \mid$ g_i *and* g_j *are compatible* $\}$*.*

Any connected component γ *of* Γ *is called a* ***movement*** *of* Γ*.*

Intuitively a concurrency graph of a go theory G groups together all atoms in G that can be achieved in one continuous movement during a single time interval. It is important to note that there may be many concurrency graphs for G - this is because E can be any subset of the set of compatible pairs of go-atoms.

Example 3 *Let* $G = \{g_1, g_2, g_3\}$ *be a go theory where* g_1 *and* g_2 *are the two compatible atoms in Example 1 and* $g_3 = go(o, (40, 60, 55), (40, 120, 55), 15, 18, 25, 28, 6, 6)$*. Then*

$\Gamma_1 = \langle G, \{(g_1, g_2)\}\rangle$, $\Gamma_2 = \langle G, \emptyset\rangle$ and $\Gamma_3 = \langle G, \{(g_1, g_2), (g_3, g_2)\}\rangle$ are all concurrency graphs of G. Furthermore Γ_1 has two movements, $\{g_1, g_2\}$ and $\{g_3\}$, whereas Γ_2 has three movements, $\{g_1\}$, $\{g_2\}$ and $\{g_3\}$. Finally, Γ_3 has only one movement $\{g_1, g_2, g_3\}$.

Definition 9 *[satisfaction of movements w.r.t. a time interval] Let G be a go theory and $\Gamma = \langle V, E\rangle$ be a concurrency graph for G. Let $\gamma = \{g_1, \ldots g_n\}$ be a movement of Γ. An interpretation I satisfies γ over a time interval $[T_1, T_2]$ iff there exist intervals $[t_{11}, t_{12}], \ldots [t_{n1}, t_{n2}]$ such that*

- *$\forall g_i \in \gamma$, I satisfies g_i over $[t_{i1}, t_{i2}]$*
- *$\forall g_i, g_j \in \gamma$ and $(g_1, g_2) \in E$, $t_{i2} > t_{j1}$ and $t_{j2} > t_{i1}$*
- *$[T_1, T_2] = [t_{11}, t_{12}], \cup \cdots \cup [t_{n1}, t_{n2}]$*

Example 4 *Let G be the go theory and Γ_1 be the concurrency graph in Example 3. The two movements of Γ_1 are $\gamma_1 = \{g_1, g_2\}$ and $\gamma_2 = \{g_3\}$. As shown in Example 1, there is an interpretation I that satisfies γ_1 over $[4, 16]$. The reader can easily verify that there is also an interpretation I' that satisfies γ_2 over $[17, 27]$.*

Definition 10 (Coherent concurrency graph) *Let G be a go theory and Γ be a concurrency graph of G. Let γ be a movement of Γ. γ is **coherent** iff there exists an interpretation I and a time interval $[T_1, T_2]$ such that $I \models \gamma$ over $[T_1, T_2]$. Γ is **coherent** iff every movement γ of Γ is coherent.*

Example 5 *Let G be the go theory and let Γ_1,Γ_2 and Γ_3 be the concurrency graphs in Example 3. As the preceding example demonstrates, both movements of Γ_1 are coherent and hence Γ_1 is coherent. It is easy to verify that Γ_2 is also coherent because each atom* *in it is coherent. However,* Γ_3 *is not coherent because there is no way to achieve all three atoms in a single movement.*

We will show later that checking consistency of a *go*-theory requires finding a coherent concurrency graph Γ of the theory. Before presenting the coherence constraints associated with a movement, we define the start point, end point, and direction of a movement.

Definition 11 (Start/end points and direction of a movement) *Let* Γ *be a concurrency graph of a go-theory* G*. Let* $\gamma = \{g_1, \ldots, g_k\}$ *be a movement of* Γ*. Suppose* $\bigcup_{i=1}^{k} LS(g_i)$ *is a single line segment* $L = [P_1, P_2]$ *and suppose* $\{p_1, \ldots, p_n\} = \{loc_1(g_i), loc_2(g_i) \,|\, 1 \leq i \leq k\}$*. Without loss of generality, assume that the points* $p_1, \ldots, p_n$ *are listed in ascending order of their distance from* P_1*. Then:*

- *The origin of* γ *is* $loc_1(\gamma) = P_1$
- *The destination of* γ *is* $loc_2(\gamma) = P_2$
- *The line segment of* γ *is* $LS(\gamma) = [P_1, P_2]$
- *The direction of* γ *is* $\vec{\gamma} = \vec{v}(g_1)$
- *The check points of* γ *in ascending order of their distance from the origin of* γ *is the set* $CheckPoints(\gamma) = [p_1, \ldots, p_n]$.

The following example illustrates the concepts defined above.

Example 6 *Let* G *be the go theory and* Γ_1 *be the concurrency graph in Example 3.* $\gamma = \{g_1, g_2\}$ *is a movement of* Γ_1*. The origin of* γ *is* $(40, 10, 55)$*. The destination of* γ *is* $(40, 90, 55)$*. The line segment of* γ *is* $LS(\gamma) = [(40, 10, 55), (40, 90, 55)]$*. The direction* *of* γ *is* $\vec{v}(\gamma) = (0, 1, 0)$. *The check points of* γ *are* $CheckPoints(\gamma) = [(40, 10, 55)$, $(40, 30, 55)$, $(40, 60, 55)$, $(40, 90, 55)]$.

We are now ready to present the concept of coherence constraints - these will be used later to derive necessary and sufficient conditions for a movement to be coherent.

Definition 12 (Coherence Constraints) *Let G be a go theory and* Γ *be a concurrency graph of G. Let* $\gamma = \{g_1, \ldots, g_k\}$ *be the movement of* Γ *and let* $CheckPoints(\gamma) = [p_1, \ldots, p_n]$. *The **coherence constraints**,* $\mathcal{L}(\gamma)$, *of* γ, *is the set:*

1. $t_1^-(g_j) \leq T_i \leq t_1^+(g_j)$ *for every* i, j *such that* $p_i = loc_1(g_j)$;
2. $t_2^-(g_j) \leq T_i \leq t_2^+(g_j)$ *for every* i, j *such that* $p_i = loc_2(g_j)$;
3. $\mathsf{dist}(p_i, p_{i+1}) \leq (T_{i+1} - T_i) \times v_i^+,\ i = 1, \ldots, n-1$;
4. $(T_{i+1} - T_i) \times v_i^- \leq \mathsf{dist}(p_i, p_{i+1}),\ i = 1, \ldots, n-1$;

where

- $T_1, \ldots, T_n$ *are variables;*
- $v_i^- = \max\{v^-(g) \mid [p_i, p_{i+1}]$ *is a subsegment of the line segment* $LS(g)\}$;
- $v_i^+ = \min\{v^+(g) \mid [p_i, p_{i+1}]$ *is a subsegment of the line segment* $LS(g)\}$.

Intuitively, T_i denotes the actual time at which the object o leaves/arrives point p_i. The first constraint above says that if g_j is any go-atom involving leaving from location p_i, then T_i must lie within the earliest departure time and the latest departure time from point p_i according to g_j. The second constraint says that if g is any go-atom that describes when o arrives at p_i then T_i must lie within the times at which o can reach p_i as well.

The third and fourth constraints say that the arrival time of object o at p_{i+1} from point p_i must be compatible with the distance between these two points and the velocity of the object. Note that if multiple go-atoms cover the line segment between p_i and p_{i+1}, then the velocities in question must all apply to the movement of o from p_i to p_{i+1}.

Note that as $\mathcal{L}(\gamma)$ only contains linear constraints, there are Linear Programming (LP) solvers [46, 45] that can solve $\mathcal{L}(\gamma)$ in polynomial time.[1]

Example 7 *Let G be the go theory and Γ_1 be the concurrency graph in Example 3. Once again, $\gamma = \{g_1, g_2\}$ is a movement of Γ_1. The set $\mathcal{L}(\gamma)$ of coherence constraints is shown below:*

$$
\begin{array}{ll}
1 \leq T_1 \leq 5 & 8 \leq T_2 \leq 10 \\
6 \leq T_3 \leq 14 & 13 \leq T_4 \leq 18 \\
\mathsf{dist}(p_1, p_2) \leq (T_2 - T_1) \times 10 & \mathsf{dist}(p_2, p_3) \leq (T_3 - T_2) \times 10 \\
\mathsf{dist}(p_3, p_4) \leq (T_4 - T_3) \times 12 & (T_2 - T_1) \times 5 \leq \mathsf{dist}(p_1, p_2) \\
(T_3 - T_2) \times 6 \leq \mathsf{dist}(p_2, p_3) & (T_4 - T_3) \times 6 \leq \mathsf{dist}(p_3, p_4)
\end{array}
$$

where $p_1 = (40, 10, 55)$, $p_2 = (40, 30, 55)$, $p_3 = (40, 60, 55)$ and $p_4 = (40, 90, 55)$. A solution to the constraints above is $T_1 = 4$, $T_2 = 8$, $T_3 = 13$, $T_4 = 16$ - hence, $\mathcal{L}(\gamma)$ is satisfiable.

The following lemma establishes a necessary and sufficient condition for a movement and concurrency graph to be coherent.

[1] Later in this chapter, we will show that an even better time bound can be achieved. $\mathcal{L}(\gamma)$ can be transformed into a Simple Temporal Problem (STP) [25], and the satisfiability of the STP can be checked in $O(n^3)$.

Lemma 2 *Let G be a go theory, Γ be a concurrency graph of G and let γ be movement of Γ. Then (i) γ is coherent iff $\mathcal{L}(\gamma)$ has a solution (ii) Γ is coherent iff for all movements γ of Γ, $\mathcal{L}(\gamma)$ is solvable.*

At various points in this thesis, we will need to refer to the specific variables in $\mathcal{L}(\gamma)$ and the speed limits in certain portions of the line segment $LS(\gamma)$ for a given movement γ. The following definition associates variables of $\mathcal{L}(\gamma)$ with checkpoints of γ and defines the speed limits over subsegments of $LS(\gamma)$.

Definition 13 (Min/max speed allowed in a movement) *Let P be a point and Γ be a concurrency graph for a go-theory G. Let $\gamma = \{g_1, \ldots, g_k\}$ be a movement of Γ such that $CheckPoints(\gamma) = [p_1, \ldots, p_n]$. Suppose $P = p_i$ for some $1 \leq i \leq n$. Then the **variable representing** P in γ, denoted by $Var(\gamma, P)$, is the variable T_i in $\mathcal{L}(\gamma)$.*

Let P_1 and P_2 be two points on the line segment $LS(\gamma)$.Then:

- *The **minimum speed allowed in** γ **on** $[P_1, P_2]$ is $v^-(\gamma, P_1, P_2) = \max\{v^-(g) \mid g \in \gamma$ and $[P_1, P_2] \cap [loc_1(g), loc_2(g)]$ is a line segment $\}$.*
- *The **maximum speed allowed in** γ **on** $[P_1, P_2]$ is $v^+(\gamma, P_1, P_2) = \min\{v^+(g) \mid g \in \gamma$ and $[P_1, P_2] \cap [loc_1(g), loc_2(g)]$ is a line segment $\}$.*

The following example illustrates the above speed bounds.

Example 8 *Let γ be the movement and $\mathcal{L}(\gamma)$ be the coherence constraints in Example 7. In this case, the checkpoints of γ are $[(40, 10, 55), (40, 30, 55), (40, 60, 55), (40, 90, 55)]$. Let $P_1 = (40, 10, 55)$, $P_2 = (40, 30, 55)$, $Q_1 = (40, 15, 55)$ and $Q_2 = (40, 50, 55)$. Then:*

- *$Var(\gamma, P_1)$ is T_1 and $Var(\gamma, P_2)$ is T_2. $Var(\gamma, Q_1)$ and $Var(\gamma, Q_2)$ are undefined because Q_1 and Q_2 are not checkpoints of γ.*
- *$v^-(\gamma, P_1, P_2) = 5$ and $v^+(\gamma, P_1, P_2) = 10$*
- *$v^-(\gamma, Q_1, Q_2) = 6$ and $v^+(\gamma, Q_1, Q_2) = 10$*

3.4 Ordering Movements and Plans

Consider a pair of go-atoms g_1, g_2. Intuitively, there are three cases in which $\{g_1, g_2\}$ is consistent:

1. if it is possible to end g_1 before g_2 starts, which can happen iff $t_2^-(g_1) < t_1^+(g_2)$;
2. if it is possible to end g_2 before g_1 starts, which can happen iff $t_2^-(g_2) < t_1^+(g_1)$;
3. if it is possible for g_1 and g_2 to overlap, which can happen iff g_1 and g_2 are compatible.

We will generalize this intuition with the help of the concurrency graph notion for arbitrary go theories.

Definition 14 (Ordering on movements) *Let G be a go theory, and let Γ be a concurrency graph for G. We define a partial order on Γ's movements as follows. $\gamma \preceq \gamma'$ iff*

$$(\forall g \in \gamma)(\ \forall g' \in \gamma')\ \ t_2^+(g) \leq t_1^-(g').$$

A total order $\sqsubseteq$ on movements of Γ is compatible with Γ iff $\sqsubseteq$ is a topological sort of $\preceq$.

We now associate a set of linear constraints with each total ordering compatible with Γ.

Definition 15 (Linear constraints of a totally ordered concurrency graph) *Suppose G is a go theory, Γ is a concurrency graph for G, and $\sqsubseteq$ is a total order compatible with Γ. Then $\mathcal{C}(G, \Gamma, \sqsubseteq)$ is set of linear constraints such that*

- *for every movement γ of Γ, $\mathcal{C}(G, \Gamma, \sqsubseteq)$ contains $\mathcal{L}(\gamma)$,*
- *for every pair γ, γ' of movements in Γ such that $\gamma \sqsubseteq \gamma'$ and $loc_2(\gamma) = loc_1(\gamma')$, $\mathcal{C}(G, \Gamma, \sqsubseteq)$ contains the constraint*

$$Var(\gamma, P) \leq Var(\gamma', Q) \text{ where } P = loc_2(\gamma) \text{ and } Q = loc_1(\gamma');$$

- *for every pair γ, γ' of movements in Γ such that $\gamma \sqsubseteq \gamma'$ and $loc_2(\gamma) \neq loc_1(\gamma')$, $\mathcal{C}(G, \Gamma, \sqsubseteq)$ contains the constraint*

$$Var(\gamma, P) < Var(\gamma', Q) \text{ where } P = loc_2(\gamma) \text{ and } Q = loc_1(\gamma');$$

Intuitively, a plan for G is a way of sequentially executing the movements in the concurrency graph of G.

Definition 16 (Plan) *Suppose G is a go theory, Γ is a concurrency graph for G and $\sqsubseteq$ is a total order compatible with Γ such that $\mathcal{C}(G, \Gamma, \sqsubseteq)$ has a solution. Then $\pi = \langle \Gamma, \sqsubseteq \rangle$ is a* ***plan*** *for G.*

A subset γ of G is a ***movement in*** *π iff γ is a movement of Γ.*

The following theorem states that a go theory is consistent iff it is possible to find a plan for it.

Theorem 2 *A go theory G is consistent iff there is a plan π for G.*

As a consequence, in order to check consistency, we need to find a way of sequencing the movements of G.

3.5 Algorithms

In this section I will present a nondeterministic polynomial time algorithm for checking consistency of go-theories. I will also demonstrate how we can implement this algorithm as an exponential-time deterministic procedure.

3.5.1 Nondeterministic Polynomial-time Algorithm

We are now ready to present the $\mathsf{Consistent}(G)$ algorithm (see Algorithm 3.1) to check consistency of an arbitrary *go*-theory. $\mathsf{Consistent}(G)$ runs in nondeterministic polynomial time.

Algorithm $\mathsf{Consistent}$(G)

1. Let $\mathcal{C}$ be a set of linear constraints that is initially empty.
2. Let Γ be a concurrency graph of G whose edges are chosen nondeterministically from the set $\{(g_i, g_j) \mid g_i$ and g_j are compatible$\}$.
3. If Γ is not coherent then return "no"
4. For every pair of movements γ, γ' of Γ, nondeterministically select $\gamma \sqsubseteq \gamma'$ or $\gamma' \sqsubseteq \gamma$
5. If $\sqsubseteq$ is not compatible with Γ then return "no".
6. For every movement γ of Γ, insert $\mathcal{L}(\gamma)$ into $\mathcal{C}$.
7. For every pair of movements $\gamma \sqsubseteq \gamma'$ of Γ do
   - If $loc_2(\gamma) = loc_1(\gamma')$ then insert the following constraint into $\mathcal{C}$

     $$Var(\gamma, P) \leq Var(\gamma', Q) \text{ where } P = loc_2(\gamma) \text{ and } Q = loc_1(\gamma')$$
   - If $loc_2(\gamma) \neq loc_1(\gamma')$ then insert the following constraint into $\mathcal{C}$

     $$Var(\gamma, P) < Var(\gamma', Q) \text{ where } P = loc_2(\gamma) \text{ and } Q = loc_1(\gamma')$$
8. If $\mathcal{C}$ has a solution then return "yes," else return "no."

Algorithm 3.1: Nondeterministic consistency checking algorithm

To see that the algorithm runs in nondeterministic polynomial time, note that in every execution trace, first 7 steps end after a polynomial number of steps, while Step 8

Algorithm CheckConsistency(G)

1. Let $O = \{(g_i, g_j) \mid g_i$ and g_j are compatible go-atoms $\}$
2. Let Γ be a graph with no edges whose vertex set is G
3. Let $C = \emptyset$ be an empty set of constraints
4. **return** SolveConstraints(O, Γ,$\emptyset$,C)

Algorithm 3.2: Deterministic consistency checking algorithm, top level procedure

can be implemented using a polynomial-time linear programming solver.

3.5.2 Deterministic Algorithm

Just as with any nondeterministic polynomial-time algorithm, Consistent(G) can be translated into a sound and complete *deterministic* algorithm that runs in exponential time. We now present the deterministic version of Consistent(G), CheckConsistency(G) (see Algorithm 3.2).

The CheckConsistency(G) algorithm calls the SolveConstraints(O, Γ,SP,C) procedure (see Algorithm 3.3), which performs a depth-first search over all possible alternatives. It is a recursive algorithm which in the first phase builds the graph Γ, then in the second phase selects an ordering over all movements of Γ and finally checks to see if the set C of constraints is solvable (complexity of this step will be discussed in detail later in this section). If at any point the algorithm returns "yes" all recursive calls return "yes" and the algorithm stops searching. On the other hand SolveConstraints(O, Γ,SP,C) returns "no" only when all choices return "no." The reader can verify that at every decision point there are only 2 choices and the depth of the recursion can be at most $O(n^2)$, thus the time complexity for CheckConsistency(G) is $O(2^{n^2})$.

Algorithm SolveConstraints(O, Γ,SP,C)

1. **if** $O \neq \emptyset$ **then**
   - (g_i, g_j)= is any element of O and $O' = O - \{(g_i, g_j)\}$
   - **if** SolveConstraints(O', Γ,SP,C) **then return** "yes"
   - **else**
     - $\Gamma' = \Gamma$ with additional edge (g_i, g_j)
     - **return** SolveConstraints(O, Γ',SP,C)
2. **elseif** $C = \emptyset$ **then**
   - Let S be the set of all movements of Γ and SP' be the set of all pairs in S
   - Let $C' = \{\mathcal{L}(s) \mid s \in S\}$
   - **return** SolveConstraints(O, Γ,SP',C')
3. **elseif** $SP = \emptyset$ **then return** if C is solvable
4. **else**
   - (s_1, s_2)=any element of SP and $SP' = SP - \{(s_1, s_2)\}$
   - $C' = C+$ constraints of s_1 before s_2
   - **if** SolveConstraints(O, Γ,SP',C') **then return** "yes"
   - **else**
     - $C' = C+$ constraints of s_1 after s_2
     - **return** SolveConstraints(O, Γ,SP',C')

Algorithm 3.3: Deterministic consistency checking algorithm

Example 9 *Let*

$$g_1 = go(o, (200, 300, 169), (200, 500, 169), 10, 20, 30, 70, 4, 10);$$

$$g_2 = go(o, (200, 400, 169), (200, 600, 169), 20, 25, 40, 60, 5, 10);$$

$$g_3 = go(o, (40, 10, 300), (40, 60, 300), 1, 5, 6, 14, 5, 10).$$

Then only g_1 and g_2 satisfy the compatibility conditions in Lemma 1.

Figure 3.1 shows the execution trace of the algorithm SolveConstraints*(O, Γ, SP, C) when it is invoked by* CheckConsistency(G) *with the parameters shown in the root node. In the left subtree,* SolveConstraints *explores the cases where g_1 and g_2 are* *not concurrent and Γ contains 3 movements. Each leaf of the left subtree corresponds to a different possible ordering of the movements; none of these leaves have solvable constraints. In the right subtree* SolveConstraints *explores the cases where g_1 and g_2 do overlap and Γ contains 2 movements. A solution exists only when the combined motion of g_1 and g_2 comes after g_3.*

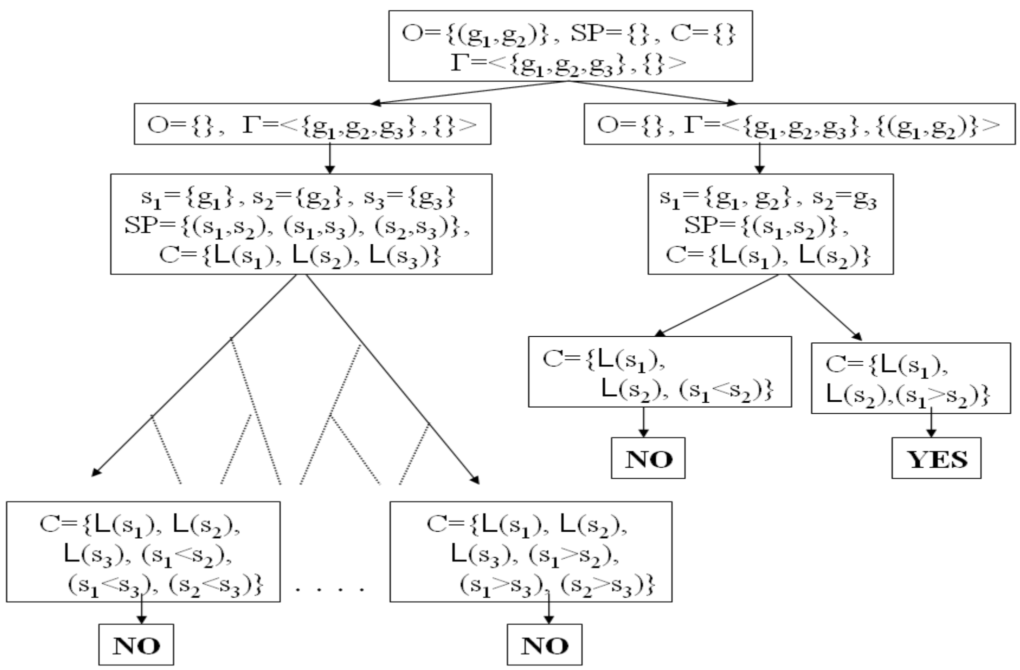


Figure 3.1: Trace of CheckConsistency(G) for Example 9

Theorem 3 *Algorithm* Consistent(G) *is correct, i.e., G is consistent iff there is a way to make the nondeterministic choices in Step 2 and 4 such that the algorithm returns "yes." Likewise, Algorithm* CheckConsistency(G) *is correct, i.e. G is consistent iff* CheckConsistency(G) *returns "yes."*

3.5.3 Complexity of the Algorithm

Checking the solvability of the set of constraints $\mathcal{C}(G, \Gamma, \sqsubseteq)$ is the last step in consistency checking. Once again we remind that all constraints in $\mathcal{C}(G, \Gamma, \sqsubseteq)$ are linear hence the system can be solved by any linear constraint solver. However we can do better by using the algorithm for solving Simple Temporal Problems [25]. A simple temporal problem (STP) is a set of temporal constraints of the form $d_{min} \leq t_i \leq d_{max}$ or $d_{min} \leq t_i - t_j \leq d_{max}$ where t_i and t_j are either constant or variables representing start or end time of an activity and d_{min}, t_i, t_j and d_{max} are reals. An STP contains at most one constraint per t_i, t_j pair. Fortunately the constraints in $\mathcal{C}(G, \Gamma, \sqsubseteq)$ are of the same form as in a STP and there is at most one constraint per variable. The next example demonstrates the types of constraints allowed in STP's and how to represent them as a directed graph used in the algorithm.

Example 10 *A ship s leaves city A sometime during the $[4, 5]$ interval and arrives at B sometime in $[15, 30]$. Then s leaves city B sometime at $[25, 50]$ and arrives at C sometime during $[40, 55]$. If it takes $[10, 20]$ unit time to go from A to B and $[15, 30]$ to go from B to C and $[5, 8]$ unit time to load s in B, is there a schedule for s that will be consistent with all these constraints*

This problem can be converted to a directed graph as in Figure 3.2. In this figure t_0 represents the beginning of time, t_1 and t_2 are leaving time from A and arriving time to B. Similarly t_3 and t_4 are the departure/arrival time from B /to C.

It is shown in [25] that existence of a temporal assignment for each t_i such that all temporal constraints are satisfied can be checked in $O(n^3)$ time where n is the number of

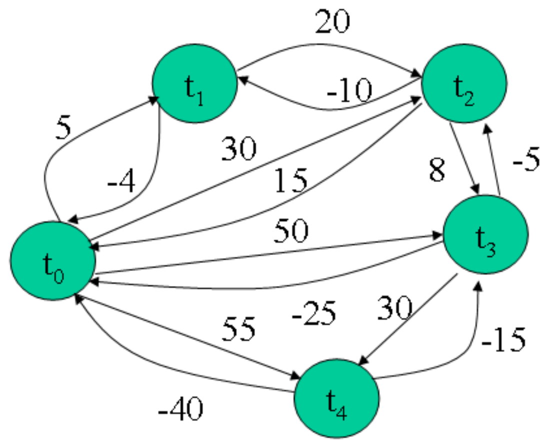


Figure 3.2: Graphical representation of STP in Example 10

temporal constraints. The algorithm creates a directed graph representing the constraints and then applies Floyd-Warshall's shortest path algorithm to detect any negative cycles in the graph. For a consistent STP, consistency checking algorithm also outputs the minimal network, which has the minimum and maximum possible values for each temporal variable.Hence a minimal solution domain for all variables in $\mathcal{C}(G, \Gamma, \sqsubseteq)$ can be computed in $O(n^3)$ time where n is the number of go atoms in G.

Chapter 4

Simple Go Theories

4.1 Definition

A class of *go*-theories called **simple *go*-theories** can be defined for which the problem of checking consistency is polynomially solvable. In addition, given an arbitrary *go*-theory G, it is possible to check whether G is simple or not in polynomial time.

The consistency check algorithm described in the preceding section makes two nondeterministic choices: (i) creating the concurrency graph Γ whose movements represent atoms to be satisfied concurrently and (ii) imposing an ordering on each pair of movements in Γ. In this section, we define a class of go-theories called *simple go-theories* for which these nondeterministic steps can be eliminated. As a result, checking consistency of simple go-theories will be polynomial. We provide the `CheckSimple` algorithm to check if a given go theory is simple, while our `CheckSimpleConsistency` algorithm checks if a simple go theory is consistent. Once again, throughout this section we assume all atoms in a go-theory are about a single object.

Definition 17 ($\mathbf{MinEnd}$ and $\mathbf{MaxStart}$) *Suppose G is a go theory and Γ is a coherent concurrency graph for G. If γ is a movement of Γ such that $P = loc_1(\gamma)$ and $Q = loc_2(\gamma)$, then*

1. $MaxStart(\gamma) = min\{t_1^+(g) \mid loc_1(g) = loc_1(\gamma)\}$

2. $MinEnd(\gamma) = max\{t_2^-(g) \mid loc_2(g) = loc_2(\gamma)\}$

Intuitively $MaxStart(\gamma)$ (resp.$MinEnd(\gamma)$) represent the latest possible start time (resp. earliest end time) for the movement represented by combination of atoms in γ.

Example 11 *Consider the go theory $G = \{g_1, g_2, g_3\}$ in example 9 where:*

$$g_1 = go(obj_1, (200, 300, 169), (200, 500, 169), 10, 20, 30, 70, 4, 10);$$

$$g_2 = go(obj_1, (200, 400, 169), (200, 600, 169), 20, 25, 40, 60, 5, 10);$$

$$g_3 = go(obj_1, (40, 10, 300), (40, 60, 300), 1, 5, 6, 14, 5, 10).$$

Let $\Gamma = \langle G, \{(g_1, g_2)\}\rangle$ be a coherent concurrency graph of G. Γ has two movements namely $\gamma_1 = \{g_3\}$ and $\gamma_2 = \{g_1, g_2\}$. The latest start time for γ_1, $MaxStart(\gamma_1)$ is 5, while the earliest end time for γ_1, $MinEnd(\gamma_1)$ is 6. For γ_2, $MaxStart(\gamma_2)$ is 20, while $MinEnd(\gamma_2)$ is 40.

Definition 18 (Necessary/maximal concurrency graph) *Let G be a go theory. The* ***necessary concurrency graph of*** *G is a concurrency graph $\Gamma^* = \langle V^*, E^*\rangle$ of G such that $(g, g') \in E^*$ iff:*

- *g and g' are compatible.*
- *It is not possible for g to finish before g', i.e. $t_2^-(g) \geq t_1^+(g')$*
- *It is not possible for g' to finish before g, i.e. $t_2^-(g') \geq t_1^+(g)$*

Furthermore Γ^* is the ***maximal concurrency graph of*** *G iff for every concurrency graph $\Gamma = \langle V, E\rangle$ of G, $E \subseteq E^*$.*

We provide an example of these concepts below.

Example 12 *Let* $G = \{g_1, g_2, g_3\}$ *be as in Example 11. Suppose* $\Gamma = \langle G, \{(g_1, g_2)\}\rangle$ *and* $\Gamma' = \langle G, \emptyset\rangle$ *are two concurrency graphs of G.* Γ *is the necessary concurrency graph of* G *because it contains all and only the edges for all atoms that has to overlap.* Γ' *on the other hand is not because it is missing the edge* (g_1, g_2)*. Furthermore* Γ *is the maximal concurrency graph of* G *since there is no pair of atoms that are compatible and can be performed separately.*

We are now ready to formally define simple go theories.

Definition 19 (Simple go-theory) *Let G be a* go-*theory and* Γ^* *be the necessary concurrency graph of* G*.* G *is a **simple go-theory** if all of the following hold:*

1. Γ^* *is the maximal concurrency graph of* G
2. *If* Γ^* *is coherent then for any two movements* γ_1, γ_2 *of* Γ^**, at most one of the following holds:*
   (a) *It is possible for* γ_1 *to finish before* γ_2*, that is* $MinEnd(\gamma_1) < MaxStart(\gamma_2)$
   (b) *It is possible for* γ_2 *to finish before* γ_1*, that is* $MinEnd(\gamma_2) < MaxStart(\gamma_1)$

4.2 Identifying Simple Go-Theories

Basically a go theory is simple if for any two atoms g_i, g_j, there is only one way to achieve them both - either concurrently or one after another. Furthermore for any two movements it is possible to decide which one comes before the other.

Example 13 *The go theory in Example 11 is a simple theory.*

The CheckSimple(G) algorithm (see Algorithm 4.4) checks if a go theory is simple.

Algorithm CheckSimple(G)

1. $E = \emptyset$
2. **for each** pair of go-atoms $g_i, g_j \in G$ **do**
   - poss = 0
   - **if** $t_2^-(g_i) < t_1^+(g_j)$ **then** poss=poss+1
   - **if** $t_2^-(g_j) < t_1^+(g_i)$ **then** poss=poss+1
   - **if** g_i, g_j satisfy compatibility conditions and $\mathcal{L}(\{g_i, g_j\})$ is solvable
     - **if** poss > 0 **then return** false **else** $E = E \cup \{g_i, g_j\}$
3. **if** $\Gamma^* = \langle G, E\rangle$ is not coherent **then return** true
4. **for each** pair of movements γ_i, γ_j of Γ^* **do**
   - poss = 0
   - **if** $MinEnd(\gamma_i) < MaxStart(\gamma_j)$ **then** poss=poss+1
   - **if** $MinEnd(\gamma_j) < MaxStart(\gamma_i)$ **then** poss=poss+1
   - **if** poss > 1 **then return** false
5. **return** true

Algorithm 4.4: CheckSimple algorithm

4.3 Consistency Checking

The following lemma states that when checking consistency of a simple go theory, it is enough to consider only the maximal concurrency graph of G.

Lemma 3 *A simple go theory G is consistent iff there exists a plan* $\pi = \langle\Gamma^*, \sqsubseteq\rangle$ *for G where* Γ^* *is the maximal concurrency graph of G.*

Lemma 3 is intuitive. The maximal concurrency graph contains all the necessary movements and for a simple go theory there are no other possible movements to consider. Lemma 3 allows us to eliminate over half the search space in example 9 because we do not have to consider the left branch which has a non maximal concurrency graph.

Definition 20 ($\sqsubseteq^*$-ordering on movements) *Let G be a simple go theory and let* Γ^* *be* *the maximal concurrency graph of* G*. Suppose* Γ^* *is coherent. Then for any movements* γ, γ' *of* Γ^**, we say that* $\gamma \sqsubseteq^* \gamma'$ *iff* $MinEnd(\gamma) < MaxStart(\gamma')$*.*

The following theorem presents a necessary and sufficient condition for a simple go theory to be consistent.

Theorem 4 *Let G be a simple go theory,* Γ^* *be the maximal concurrency graph of G and* $\sqsubseteq^*$ *be the ordering in Definition 20. G is consistent iff*

- Γ^* *is coherent, and*
- $\sqsubseteq^*$ *is a total order, and*
- $\mathcal{C}(G, \Gamma^*, \sqsubseteq^*)$ *has a solution*

Algorithm CheckSimpleConsistency(G) (see Algorithm 4.5) checks the consistency of simple *go*-theories. In the first step, the algorithm builds the minimal concurrency graph - if it is not coherent, it returns false. It then checks if $\sqsubseteq^*$ is a total order - if not, it returns false. Finally, if $\mathcal{C}(G, \Gamma^*, \sqsubseteq^*)$ has a solution, CheckSimpleConsistency(G) returns true - otherwise it returns false.

4.4 Computational Properties of Simple Go-Theories

Other than polynomial time consistency checking, for simple go-theories answering entailment queries can be done in polynomial time. The section presents the theoretical reasoning leading to this result.

First we are going to define a model being an instance of a plan and then equality of plans. Using this definition we will show that all plans for a simple theory are equal.

Algorithm $\mathsf{CheckSimpleConsistency}(G)$

1. Let $C = \{\}$ and Γ^* be the maximal concurrency graph of G
2. **for each** movement γ of Γ^* **do**
   - **if** $\mathcal{L}(\gamma)$ has no solution **then return** false **else** $C = C \cup \mathcal{L}(\gamma)$
3. **if** $\exists \gamma, \gamma'$ such that neither $\gamma \sqsubseteq^* \gamma'$ or $\gamma' \sqsubseteq^* \gamma$ **then return** false
4. **for each** movement pair $\gamma \sqsubseteq^* \gamma'$ of Γ^* **do**
   - $P = loc_2(\gamma)$ and $Q = loc_1(\gamma')$
   - **if** $P = Q$ **then** $C = C \cup \{Var(\gamma, P) \leq Var(\gamma', Q)\}$
   - **else** $C = C \cup \{Var(\gamma, P) < Var(\gamma', Q)\}$
5. **if** C has a solution **then return** true
6. **else return** false

Algorithm 4.5: CheckSimpleConsistency algorithm

Definition 21 *[satisfaction of a movement w.r.t. a plan, instance of a plan] Let G be a go theory and* $\pi = \langle \Gamma, \sqsubseteq \rangle$ *be a plan for G. Let* $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ *be the movements in* π. *An interpretation* I *is an* ***instance*** *of* π *iff there are time intervals* $[t_{11}, t_{12}], \ldots [t_{n1}, t_{n2}]$ *such that*

- I *satisfies* γ_i *over* $[t_{i1}, t_{i2}]$ *and*
- $\forall \gamma_i \sqsubseteq \gamma_j, \quad t_{i2} < t_{j1}$

Furthermore I ***satisfies*** γ_i ***w.r.t.*** π *over* $[t_{i1}, t_{i2}]$

Basically an interpretation is an instance of a plan iff it satisfies the atoms in the theory as directed by the plan.

Definition 22 *Let G be a consistent go theory. Two plans for G* π *and* π' *are equal iff*

- $\forall I$ *such that* I *is an instance of* π, I *is an instance of* π'
- $\forall I$ *such that* I *is an instance of* π', I *is an instance of* π

Theorem 5 *Suppose G is a consistent simple go theory. Then all plans for G are equal to the plan* $\pi = \langle \Gamma^*, \sqsubseteq^* \rangle$ *where* Γ^* *is the maximal concurrency graph of G and* $\sqsubseteq^*$ *is the total order given in definition 20.*

We call π *the* ***main plan*** *of G.*

This is a powerful theorem. Using this result we can answer entailment queries in polynomial time as well. This is because it as enough to check if all instances of the main plan satisfy a LOM formula. The following theorem states this result.

Theorem 6 *Suppose G is a consistent simple go theory,* π *is the main plan for* G *and* F *is a LOM formula.* $G \models F$ *iff all instances of* π *satisfy* F.

Chapter 5

Temporal, Positional and Speed Certainty Intervals

This chapter, defines the concepts of temporal, positional and speed certainty intervals for an object given a consistent go-theory G. Intuitively, the temporal certainty interval is a time interval when we are sure about an object being within a certain region, while a positional certainty region is a segment of a movement where the object is guaranteed to be at a given time. This chapter also provides lemmas showing how to compute these concepts effectively. Certainty intervals are useful for pruning when answering queries.

5.1 Temporal Certainty Interval

We now declaratively define the earliest and latest times when a given object can be at a given location.

Definition 23 (Earliest/latest arrival time at a point) *Let P be a point, G be a go theory, o be an object, and π be a plan for G^o. Suppose γ is a movement in π such that $P \in LS(\gamma)$. Then* ***o's earliest arrival time at point*** *P with respect to γ and π, denoted by $T^-(G^o, \pi, \gamma, P)$, is $\mathbf{min}\{t \mid I(o,t) = P$ and I satisfies γ w.r.t. π over a time interval that contains $t\}$.*

Similarly ***o's latest arrival time at point*** *P with respect to γ and π, denoted by $T^+(G^o, \pi, \gamma, P))$, is $\mathbf{max}\{t \mid I(o,t) = P$ and I satisfies γ w.r.t. π over a time interval that contains $t\}$.*

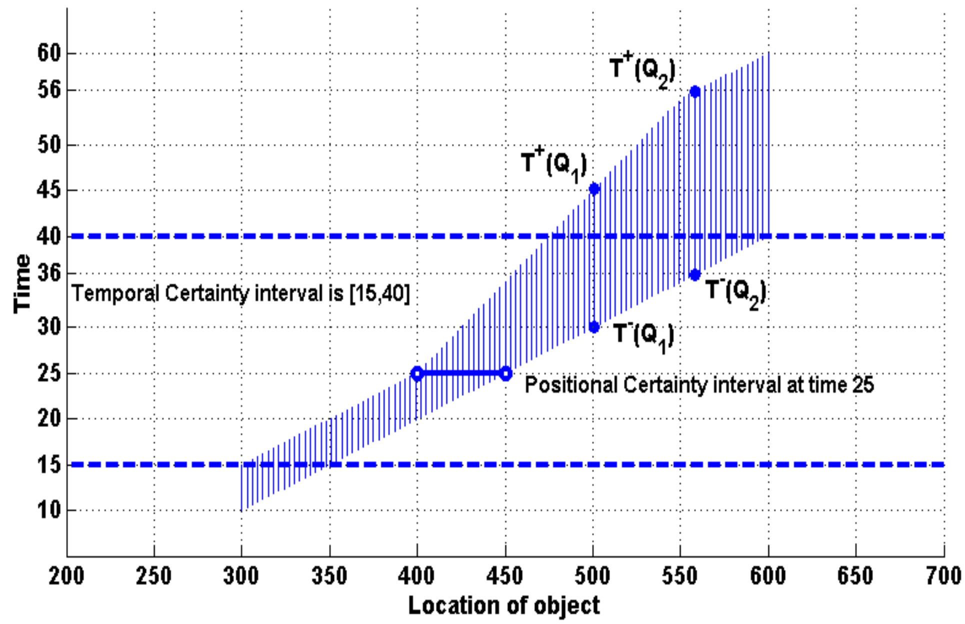


Figure 5.1: Minimum and maximum arrival times for every point on $LS(\gamma)$ in Example 14. Since $LS(\gamma)$ changes only in one dimension the constant dimensions are omitted. $T^{-}(Q_1)$ and $T^{+}(Q_1)$ are the minimum and maximum arrival time for point Q_1 in Example 14, similarly $T^{-}(Q_2)$ and $T^{+}(Q_2)$ are for point Q_2. Positional certainty region at time 25 as discussed in Example 16.

Example 14 *Let G be the go theory in Example 9. Let π be the plan for G such that $\{g_3\} \sqsubseteq \{g_1, g_2\}$. Let $\gamma = \{g_1, g_2\}$ be a movement in π. Figure 5.1 shows $T^{-}(G, \pi, \gamma, P)$ and $T^{+}(G, \pi, \gamma, P)$ for every point P on the line segment $LS(\gamma)$. For simplicity, we only display the y-coordinate of the points - the x, y coordinates do not change.*

Suppose $Q_1 = (200, 500, 169)$ and $Q_2 = (200, 560, 169)$ are two points in $LS(\gamma)$. An inspection of the figure reveals that:

- $T^{-}(G^o, \pi, \gamma, Q_1) = 30$ *and* $T^{+}(G^o, \pi, \gamma, Q_2) = 45$

- $T^-(G^o, \pi, \gamma, Q_2) = 36$ *and* $T^+(G^o, \pi, \gamma, Q_2) = 56$

The following lemmas show how we can use linear programming to compute the earliest and latest arrival times of an object w.r.t. a plan and a movement.

Lemma 4 *Let P be a point, G be a go theory, o be an object, and π be a plan for G^o. Let γ be a movement in π such that $P \in LS(\gamma)$ and $CheckPoints(\gamma) = [p_1 \dots p_n]$.*

1. ***If*** *$P = p_i$ for some i, then $T^-(G^o, \Gamma, \sqsubseteq, \gamma, P)$ is the solution of the linear programming problem:* ***minimize*** *$Var(\gamma, P)$* ***subject to*** *$\mathcal{C}(\pi)$*

2. ***If*** *the previous case does not apply and P is on line segment $[p_i, p_{i+1}]$ for some i, then $T^-(G^o, \pi, \gamma, P)$ is the* ***maximum*** *of:*

   - $T^-(G^o, \pi, \gamma, p_i) + dist(P, p_i)/v^+(\gamma, p_i, p_{i+1})$

   - $T^-(G^o, \pi, \gamma, p_{i+1}) - dist(P, p_{i+1})/v^-(\gamma, p_i, p_{i+1})$

The intuition for the second case is that the earliest arrival time at P occurs when we start as early as possible from p_i, i.e. at $T^-(G^o, \pi, \gamma, p_i)$, and move as fast as possible, i.e. at $v^+(\gamma, p_i, p_{i+1})$. However, this may not be a solution because the rest of the movement needs to satisfy the arrival time constraints at p_{i+1}. The earliest the object can arrive at p_{i+1} is at $T^-(G^o, \pi, \gamma, p_{i+1})$ - travel between P and p_{i+1} takes the most time when it is traveresd at the slowest possible speed $v^-(\gamma, p_i, p_{i+1})$.

Similarly we can compute $T^+(G^o, \pi, \gamma, P)$ which gives the latest arrival time for the object at the given point P.

Lemma 5 ($T^+(G^o, \pi, \gamma, P)$) *Let P, G, o, π and γ be as defined in Lemma 4. Suppose $CheckPoints(\gamma) = [p_1 \dots p_n]$.*

1. ***If*** $P = p_i$ *for some* i*, then* $T^+(G^o, \pi, \gamma, P)$ *is the solution of the following linear program:* ***maximize*** $Var(\gamma, P)$ ***subject to*** $\mathcal{C}(\pi)$

2. ***If*** *the previous case does not apply and* P *is on line segment* $[p_i, p_{i+1}]$ *for some* i*, then* $T^+(G^o, \pi, \gamma, P)$ *is the* ***minimum*** *of the following two items:*

   - $T^+(G^o, \pi, \gamma, p_i) + dist(P, p_i)/v^-(\gamma, p_i, p_{i+1})$

   - $T^+(G^o, \pi, \gamma, p_{i+1}) - dist(P, p_{i+1})/v^+(\gamma, p_i, p_{i+1})$

We are now ready to define the temporal certainty interval of an object w.r.t. a plan and a movement.

Definition 24 ($TCI(G^o, \pi, \gamma)$**)** *Let* G *be a go theory,* o *be an object, and* π *be a plan for* G^o*. The* ***temporal certainty interval****,* $TCI(G^o, \pi, \gamma)$*, of a movement* γ *in* π *is the time interval* $[T_1, T_2]$ *where*

- T_1 *is* $T^+(G^o, \pi, \gamma, loc_1(\gamma))$
- T_2 *is* $T^-(G^o, \pi, \gamma, loc_2(\gamma))$.

$TCI(G^o, \pi, \gamma)$ *is undefined if* $T_1 > T_2$.

Intuitively, $TCI(G^o, \pi, \gamma)$ is the interval when we know for sure that object o is within the line specified in movement γ. When $G = \{g\}$ contains a single go atom we will use $T^-(g, P)$, $T^+(g, P)$ and $TCI^+(g)$ as short hand notations.

Example 15 *Let* G*,*π *and* γ *be as in the example 14. Then* $TCI(G, \pi, \gamma) = [15, 40]$*. Figure 5.1 also depicts the temporal certainty interval of* γ.

5.2 Positional Certainty Region

We have thus far studied the problem of when a given object will be at a given point. Conversely, given a time point t and a movement γ, we may wish to know the potential segment of $LS(\gamma)$ where the object associated with γ could possibly be. To find this, we need another set of definitions.

Definition 25 (Min/max point of advancement; positional certainty region) *Let G be a go theory, o be an object, and π be a plan for G^o. Let γ be a movement in π such that $TCI(G^o, \pi, \gamma)$ is defined and t be any time point in $TCI(G^o, \pi, \gamma)$. Then* ***o's minimum point of advancement*** *with respect to γ, and π, denoted $P^-(G^o, \pi, \gamma, t)$, is the closest point P to $loc_1(\gamma)$ such that $I(o, t) = P$ where I satisfies γ w.r.t. π over a time interval including t.*

Similarly ***o's maximum point of advancement*** *with respect to γ, and π, denoted by $P^+(G^o, \pi, \gamma, t)$, is the furthest point P to $loc_1(\gamma)$ such that $I(o, t) = P$ where I satisfies γ w.r.t. π over a time interval including t.*

The ***positional certainty region****, $PCR(G^o, \pi, \gamma, t)$, on γ w.r.t. π and t is the line segment from $P^-(G^o, \pi, \gamma, t)$ to $P^+(G^o, \pi, \gamma, t)$*

When $G = \{g\}$ contains a single go atom we will use $P^-(g, t), P^+(g, t), PCR(g, t)$ as short hand notations.

Example 16 *Let G,π and γ be as in example 14. Let $t = 25$ be a time point in $TCI(G, \pi, \gamma) = [15, 40]$. By consulting Figure 14, we can see that:*

- $P^-(G, \pi, \gamma, t) = (200, 400, 169)$

- $P^+(G, \pi, \gamma, t) = (200, 450, 169)$
- $PCR(G, \pi, \gamma, 25) = [(200, 400, 169), (200, 450, 169)]$

The following lemma demonstrates how to compute $P^-(G^o, \pi, \gamma, t)$ and $P^+(G^o, \pi, \gamma, t)$. It uses the fact that time and position act as inverses of each other during the temporal certainty interval of a movement.

Lemma 6 *Let G be a go theory, o be an object, and π be a plan for G^o. Let γ be a movement in π such that $TCI(G^o, \pi, \gamma)$ is defined and let t be any time point in $TCI(G^o, \pi, \gamma)$. Then*

- *$P^-(G^o, \pi, \gamma, t) = P$ such that $T^+(G^o, \pi, \gamma, P) = t$.*
- *$P^+(G^o, \pi, \gamma, t) = P$, such that $T^-(G^o, \pi, \gamma, P) = t$.*

Lemma 7 *Let G be a go theory, o be an object, and π be a plan for G^o. Let γ be a movement in π such that $TCI(G^o, \pi, \gamma)$ is defined and let t be any time point in $TCI(G^o, \pi, \gamma)$. If I is an instance of π, then $I(o, t)$ is on the line segment $PCR(G^o, \pi, \gamma, t)$.*

5.3 Speed Interval

Similar to the earliest and latest times we can define maximum and minimum speed of an object during a movement.[1]

Definition 26 (Min/max speed on a line segment) *Let $[P_1, P_2]$ be a line segment, G be a go theory, o be an object, and π be a plan for G^o. Suppose γ is a movement in π such*

[1]This is different than the max/min speed allowed which only take into account the speed limits given in the atoms.

*that $[P_1, P_2] \in LS(\gamma)$. Then **o's maximum speed on** $[P_1, P_2]$ with respect to γ and π, denoted $V^+(G^o, \pi, \gamma, P_1, P_2)$ is the maximum value of $dI(o,t)/dt$ over the time interval $[T_1, T_2]$ where $I(o, T_1) = P_1$ and $I(o, T_2) = P_2$ I satisfies γ w.r.t. π over a time interval that contains $[T_1, T_2]\}$.*

*Similarly **o's minimum speed on** $[P_1, P_2]$ with respect to γ and π, denoted by $V^-(G^o, \pi, \gamma, P_1, P_2)$, is the minimum value of $dI(o,t)/dt$ over the time interval $[T_1, T_2]$ where $I(o, T_1) = P_1$ and $I(o, T_2) = P_2$ I satisfies γ w.r.t. π over a time interval that contains $[T_1, T_2]\}$.*

We will use the short hand notations $V^+(G, P_1, P_2)$ and $V^-(G, P_1, P_2)$ when G has only one atom.

At first this definition might seem redundant however while answering queries (specifically go-queries) we need the speed interval to be as tight as possible. Although pruning the speed interval given in the atom is not needed as frequent as pruning the time intervals, it does happen under certain conditions. We'll try to show this with a simple example and then present a lemma that shows when maximum/minimum speed are different than max/min speed allowed.

Example 17 *Let $g_1 = \{go(o, (0, 0, 0), (50, 0, 0), 3, 3, 13, 13, 5, 10)\}$, $g_2 = \{go(o, (0, 0, 0), (50, 0, 0), 3, 3, 8, 8, 5, 10)\}$ and $g_1 = \{go(o, (0, 0, 0), (50, 0, 0), 3, 3, 9, 9, 5, 10)\}$ be three theories. All tree move on X-axis for 50 units and all three allow a minimum speed 5 and maximum speed 10. However min/max speed on [(0,0,0),(50,0,0)] differs for each of them:*

- $V^+(g_1, (0, 0, 0), (50, 0, 0)) = 5$ *and* $V^-(g_1, (0, 0, 0), (50, 0, 0)) = 5$ *because all* *models of g_1 has to move with an average speed of 5 between times 3 and 13 and given the speed limits this can only be achieved by a constant speed of 5.*

- $V^+(g_2,(0,0,0),(50,0,0)) = 10$ *and* $V^-(g_2,(0,0,0),(50,0,0)) = 10$ *because all models of g_2 has to move with an average speed of 10 between times 3 and 8 and given the speed limits this can only be achieved by a constant speed of 10.*

- $V^+(g_3,(0,0,0),(50,0,0)) = 10$ *and* $V^-(g_1,(0,0,0),(50,0,0)) = 5$*. This is different from the previous two cases. We need an average speed of 8.33 during times 3 and 9 which does not enforce any constant speed constraint on its models.*

Lemma 8 *Let π be a plan for a go-theory G. Let γ be a movement of π such that $CheckPoints(\gamma) = [p_1,\ldots,p_n]$. Suppose $[P,Q]$ is a line segment such that $[P,Q] \subseteq LS(\gamma)$. Then,*

$$V^+(G^o,\pi,\gamma,P,Q) = max\{V^+(G^o,\pi,\gamma,p_i,p_{i+1}) \mid [P,Q] \cap [p_i,p_{i+1}] \text{ is line segment } \}$$

where $V^+(G^o,\pi,\gamma,p_i,p_{i+1})$ is equal to the following:

- $V^+(G^o,\pi,\gamma,p_i,p_{i+1})$=$v^-(\gamma,P,Q)$ *when*
  - $T^-(G^o,\pi,\gamma,p_i) = T^+(G^o,\pi,\gamma,p_i)$ *and*
  - $T^-(G^o,\pi,\gamma,p_{i+1}) = T^+(G^o,\pi,\gamma,p_{i+1})$ *and*
  - $v^-(\gamma,p_i,p_{i+1}) = \mathsf{dist}(p_i,p_{i+1})/(T^+(G^o,\pi,\gamma,p_{i+1}) - T^+(G^o,\pi,\gamma,p_i))$
- $V^+(G^o,\pi,\gamma,p_i,p_{i+1})$=$v^+(\gamma,P,Q)$ *otherwise*

Similarly we can compute $V^-(G^o,\pi,\gamma,P,Q)$ using the following lemma:

Lemma 9 *Let π be a plan for a go-theory G. Let γ be a movement of π such that $CheckPoints(\gamma) = [p_1, \ldots, p_n]$. Suppose $[P, Q]$ is a line segment such that $[P, Q] \subseteq LS(\gamma)$. Then,*

$$V^-(G^o, \pi, \gamma, P, Q) = max\{V^-(G^o, \pi, \gamma, p_i, p_{i+1}) \mid [P, Q] \cap [p_i, p_{i+1}] \textit{ is line segment } \}$$

where $V^-(G^o, \pi, \gamma, p_i, p_{i+1})$ is equal to the following:

- $V^-(G^o, \pi, \gamma, p_i, p_{i+1})$=$v^+(\gamma, P, Q)$ *when*
  - $T^-(G^o, \pi, \gamma, p_i) = T^+(G^o, \pi, \gamma, p_i)$ *and*
  - $T^-(G^o, \pi, \gamma, p_{i+1}) = T^+(G^o, \pi, \gamma, p_{i+1})$ *and*
  - $v^+(\gamma, p_i, p_{i+1}) = \mathsf{dist}(p_i, p_{i+1})/(T^+(G^o, \pi, \gamma, p_{i+1}) - T^+(G^o, \pi, \gamma, p_i))$
- $V^-(G^o, \pi, \gamma, p_i, p_{i+1})$=$v^-(\gamma, P, Q)$ *otherwise*

Chapter 6

Answering Ground Atomic Queries

In this section we present algorithms to answer ground *atomic* queries - answering conjunctive ground queries is a straightforward combination of the answers to ground queries and hence we do not discuss that here.

6.1 Ground Atomic go Queries

In this section, we show how to check whether a atom $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ is a logical consequence of a *go*-theory G. Without loss of generality we assume g is a normalized go-atom. Given a concurrency graph, Γ, of G, we will first identify the movements that are the spatially and temporally relevant to g.

Definition 27 *Let o be an object,* $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *be a go-atom, and* G *be a go-theory. Suppose* Γ *is a concurrency graph of* G^o*. A movement* γ *of* Γ *is* ***related*** *to* g *if:*

- γ *is* spatially relevant *to* g*, i.e. if:*
  - $[P_1, P_2]$ *is a subsegment of* $LS(\gamma)$ *and*
  - $\vec{v}(\gamma)$ =$\vec{v}(g)$*, i.e. they both have the same direction, and*
- γ *is* temporally relevant *to* g *if:*
  - $\exists g' \in \gamma \mid t_2^+(g') > t_1^-(g)$*,i.e. not all the atoms in* γ *end before* g *and*

– $\exists g' \in \gamma \mid t_1^-(g') \leq t_2^+(g)$,*i.e. not all the atoms in γ start after g.*

Given a plan π for G, we now identify some necessary condition for G to entail g. The idea is to ensure that there is a movement γ in π relevant to g such that when achieving γ, o always arrives at P_1 sometime in $[t_1^-, t_1^+]$ and at P_2 sometime in $[t_2^-, t_2^+]$, and that o's speed is in the range $[v^-, v^+]$. The following lemma formally states these conditions.

Lemma 10 *Let o be an object, $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a go-atom and G be a go-theory. Suppose π is a plan for G^o. All instances of π satisfy g iff all the following conditions hold:*

1. *There is a movement γ in π that is relevant to g*
2. $t_1^- \leq T^-(G^o, \pi, \gamma, P_1) \leq T^+(G^o, \pi, \gamma, P_1) \leq t_1^+$.
3. $t_2^- \leq T^-(G^o, \pi, \gamma, P_2) \leq T^+(G^o, \pi, \gamma, P_2) \leq t_2^+$
4. $v^- \leq V^-(\gamma, P_1, P_2) \leq V^+(\gamma, P_1, P_2) \leq v^+$.

We are now ready to define the $\mathsf{CheckGo}$ algorithm (see Algorithm 6.6) to check if all instances of a plan for a go theory entails a ground go atom.

Theorem 7 *Let G be a consistent* go-*theory and $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a ground* go *atom. Then: g is a logical consequence of G iff for every plan π for G^o* $\mathsf{CheckGo}(G, \Gamma, \sqsubseteq, g)$ *returns "true".*

The above theorem tells us that in order check if $G \models g$, we must execute the $\mathsf{CheckGo}$ algorithm for each plan π.

Algorithm $\mathsf{CheckGo}(G, \pi, g)$

1. Suppose $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$
2. **if** $C(\pi)$ has no solution **then return** true
3. **for each** movement $\gamma \in \pi$ **do**
   - **if** γ is not relevant to g **then continue**
   - **elseif** $t_1^- \leq T^-(G^o, \pi, \gamma, P_1) \leq T^+(G^o, \pi, \gamma, P_1) \leq t_1^+$ and $t_2^- \leq T^-(G^o, \pi, \gamma, P_2) \leq T^+(G^o, \pi, \gamma, P_2) \leq t_2^+$ and $v^- \leq V^-(\gamma, P_1, P_2) \leq V^+(\gamma, P_1, P_2) \leq v^+$ **then return** true
   - **else continue**
4. **return** false

Algorithm 6.6: CheckGo algorithm

Example 18 *Let G be the following simple go theory:*

$$g_1 = go(obj_1, (200, 300, 169), (200, 500, 169), 10, 20, 30, 70, 4, 10);$$

$$g_2 = go(obj_1, (200, 400, 169), (200, 600, 169), 20, 25, 40, 65, 5, 10);$$

$$g_3 = go(obj_1, (40, 10, 300), (40, 60, 300), 1, 5, 6, 14, 5, 10).$$

Let $P_1 = (200, 400, 169)$ and $P_2 = (200, 500, 169)$. obj_1 will arrive at P_1 sometime during $[20, 25]$ and it will arrive at P_2 sometime during $[30, 45]$ (subject to consistency constraints). Furthermore the minimum and maximum speeds between P_1 and P_2 are 5 and 10 respectively. Consider the atoms $q_1 = go(obj_1, (200, 400, 169), (200, 500, 169), 10, 25, 30, 45, 4, 15)$ and $q_2 = go(obj_1, (200, 400, 169), (200, 500, 169), 10, 30, 20, 46.67, 6, 20)$. q_1 is a logical consequence of G because its temporal and speed constraints are satisfied by each model of G:

- *Departure constraint:* $10 \leq 20 \leq 25 \leq 25$
- *Arrival Constraint:* $30 \leq 30 \leq 45 \leq 45$
- *Speed Constraint:* $4 \leq 5 \leq 10 \leq 15$

On the other hand, q_2 is not a logical consequence of G because of its higher minimum speed. For example if obj_1 arrives at P_1 at time 20 and travels with a constant speed of 5, then it will arrive at P_2 at time 40, thus satisfying both g_1 and g_2. However it will not satisfy q_2 because q_2 requires the speed of obj_1 to be at least 6.

6.2 Ground Atomic in Queries

In this section, we show how to check whether an atom of the form $a = \mathsf{in}(o, Q_1, Q_2, t_1, t_2)$ is a logical consequence of a *go*-theory G. First we define the cuboid volume that a ground $\mathsf{in}()$ atom represents.

Definition 28 *Let $a = \mathsf{in}(o, Q_1, Q_2, t_1, t_2)$ be a ground atom. The **volume** a **represents**, denoted $Vol(a)$, is the set of points P such that*

$$Q_1^x \leq P^x \leq Q_2^x \text{ and } Q_1^y \leq P^y \leq Q_2^y \text{ and } Q_1^z \leq P^z \leq Q_2^z$$

Consider the simple case when G has a single atom g (later, we will show how to deal with arbitrary G's). The following lemma specifies necessary and sufficient conditions for entailment of a ground $\mathsf{in}()$-atom.

Lemma 11 *Suppose $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^+, t_2^+, v^-, v^+)$ is a* go*-atom, $G = \{g\}$ is a* go*-theory and $a = \mathsf{in}(o, Q_1, Q_2, t_1, t_2)$ is a ground $\mathsf{in}()$-atom. a is a logical consequence of G iff:*

- *$Vol(a)$ intersects the line segment $LS(g)$ and*
- *o's latest arrival time at point P_1' is less than or equal to t_2 and*
- *o's earliest arrival time at point P_2' greater than or equal to t_1.*

Here, P_1' and P_2' are points on $LS(g)$ such that $L = [P_1', P_2']$ is the longest sub-segment of $LS(g)$ inside $Vol(a)$ and $\mathsf{dist}(loc_1(g), P_1') \leq \mathsf{dist}(loc_1(g), P_2')$.

The lemma above says that $G \models a$ if and only if the object o is guaranteed to be inside $Vol(a)$ at some time in the interval $[t_1, t_2]$. This intuition can therefore be used to check if a is a logical consequence of an arbitrary *go*-theory. For an arbitrary *go*-theory G and a given a concurrency graph of G, there may be more than one movement that might be of interest for a ground $\mathsf{in}()$ query. We first define the movements that are related to a ground $\mathsf{in}()$ atom.

Definition 29 (Movements related to a `in` atom) *Suppose G is a go-theory, o is an object, Γ is a concurrency graph of G^o and γ is a movement of Γ.*

1. *The **extent** of γ is given by the interval $[\min\{t_1^-(g) \mid g \in \gamma\}, \max\{t_2^+(g) \mid g \in \gamma\}]$.*
2. *γ **is related to** $a = \mathsf{in}(o, P_1, P_2, t_1, t_2)$ iff*
   - *$LS(\gamma) \cap Vol(a) \neq \emptyset$ and*
   - *$extent(\gamma) \cap [t_1, t_2] \neq \emptyset$.*

For an arbitrary *go*-theory G and an object o, we need to consider all plans for G^o because such plans determine the positional and temporal intervals for the object. The following lemma specifies necessary conditions for a go theory to entail a ground $\mathsf{in}()$-atom with respect to a specific plan.

Lemma 12 *Suppose $a = \mathsf{in}(o, Q_1, Q_2, t_1, t_2)$ is a ground $\mathsf{in}()$-atom and G is a* go*-theory, and π is a plan for G^o. All instances of π satisfy a iff there is a movement γ in π such that all the following conditions hold:*

Algorithm $\mathsf{CheckIn}(G, o, \pi, a)$

1. Suppose $a = \mathsf{in}(o, Q_1, Q_2, t_1, t_2)$;
2. Suppose $\pi = \langle \Gamma, \sqsubseteq \rangle$;
3. **if** $\mathcal{C}(\pi)$ has no solution **then return** true
4. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements in π
5. **for** i **from** 1 **to** n **do**
   - **if** γ_i is not related to a **then continue**
   - Suppose $[P_1, P_2]$= $LS(\gamma_i) \cap Vol(a)$ and $\mathsf{dist}(loc_1(\gamma_i), P_1) \leq \mathsf{dist}(loc_1(\gamma_i), P_2)$
   - **if** $T^+(G^o, \pi, \gamma_i, P_1) > t_2$ **then return** false
   - **elseif** $T^-(G^o, \pi, \gamma_i, P_2) \geq t_1$ **then return** true
   - **else continue**
6. **return** false

Algorithm 6.7: CheckIn algorithm

- *γ is related to a*
- $T^+(G^o, \pi, \gamma, P_1) \leq t_2$
- $T^-(G^o, \pi, \gamma, P_2) \geq t_1$

Where P_1 and P_2 are points on $LS(\gamma)$ such that $L = [P_1, P_2]$ is the longest subsegment of $LS(\gamma)$ inside $Vol(a)$ and $\mathsf{dist}(loc_1(\gamma), P_1) \leq \mathsf{dist}(loc_1(\gamma), P_2)$.

Algorithm CheckIn (see Algorithm 6.7) uses the above lemma directly.

Theorem 8 *Suppose G is a consistent* go-*theory and $a = \mathsf{in}(o, P_1, P_2, t_1, t_2)$ is a ground atom. Then: a is a logical consequence of G iff for every plan π for G^o algorithm* $\mathsf{CheckIn}(G, o, \pi, a)$ *returns "true".*

The above theorem implies that we merely need to run CheckIn on each plan in order to check for entailment of a ground in atom.

Example 19 *Suppose we have a go theory G and an object o such that G^o=$\{g_1, g_2, g_3\}$. Suppose $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$. Figure 6.1 depicts the volume $Vol(a)$ and three lines ℓ_1, ℓ_2, ℓ_3 representing $LS(g_1)$, $LS(g_2)$ and $LS(g_3)$ respectively. Let $\pi = \langle \Gamma, \sqsubseteq \rangle$ be a plan for G^o such that $\Gamma = < \{g_1, g_2, g_3\}, \emptyset >$ and $\{g_1\} \sqsubseteq \{g_2\} \sqsubseteq \{g_3\}$. Suppose $[t_1, t_2]$ is wide enough so that both g_1 and g_3 are relevant to a. then the algorithm* CheckIn *performs the following steps:*

- *o's latest arrival time at P_1 after t_2: then* CheckIn *returns NO*
- *o's latest arrival time at P_1 before t_2 and*
  - *o's earliest arrival time at P_2 after t_1: then* CheckIn *returns YES*
  - *o's earliest arrival time at P_2 before t_1 and*
    - *o's latest arrival time at P_3 after t_2: then* CheckIn *returns NO*
    - *o's latest arrival time at P_3 before t_2 and*
      - *o's earliest arrival time at P_4 after t_1: then* CheckIn *returns YES*
      - *o's earliest arrival time at P_4 before t_1: then* CheckIn *returns NO*

6.3 Ground Atomic near Queries

We are going to explain how to answer near queries for cases with increasing complexity. We first consider the case when there are only two atoms per object and the near atom defines a single time point instead of an interval. We then explain how to generalize the idea for arbitrary near atoms with intervals. Finally we present the case where the go theory may contain more than one atom per object and we have arbitrary near atoms.

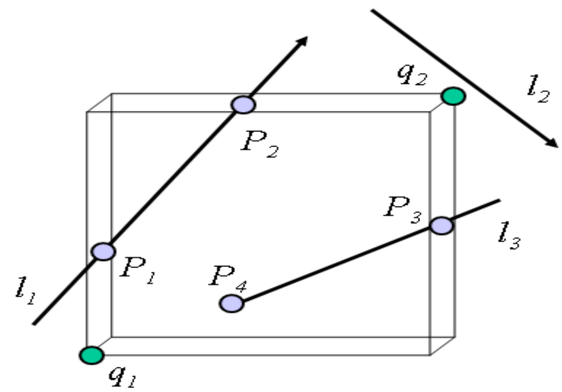


Figure 6.1: Graphical representation of G and $Vol(a)$ in Example 19

6.3.1 Binary go theories and a single time point ground near atom.

In this case we have a *go*-theory G containing two go-atoms g_1 and g_2 for objects o_1 and o_2 respectively, and we want to know if the distance between o_1 and o_2 is at most d at a given time t. Let $b = \mathsf{near}(o_1, o_2, d, t, t)$. The following lemma states necessary and sufficient conditions for G to entail b.

Lemma 13 *Let $G = \{g_1, g_2\}$ be a go theory such that $obj(g_1) = o_1$ and $obj(g_2) = o_2$. $G \models \mathsf{near}(o_1, o_2, d, t, t)$ iff:*

1. *$TCI(g_1)$ and $TCI(g_2)$ are defined and*
2. *$t \in TCI(g_1)$ and $t \in TCI(g_2)$ and*
3. *$\mathsf{dist}(P^-(g_1, t), P^-(g_2, t)) \leq d$ and*
4. *$\mathsf{dist}(P^-(g_1, t), P^+(g_2, t)) \leq d$ and*
5. *$\mathsf{dist}(P^+(g_1, t), P^-(g_2, t)) \leq d$ and*
6. *$\mathsf{dist}(P^+(g_1, t), P^+(g_2, t)) \leq d$.*

Intuitively we cannot be sure of the positions of the two objects unless t is the temporal certainty interval of both objects. Furthermore the distance between any two points in the positional certainty regions of o_1 and o_2 at time t should be less than or equal to d. Note that the positional certainty region of an object at a given time is a line segment and the maximum distance between two line segments is achieved at one of the end points of the line segments. Items (3)–(6) in Lemma 13 ensure that the maximum distance between positional certainty regions of objects is less than or equal to d.

6.3.2 Binary go theories and an arbitrary near atom

The following lemma shows how to generalize the preceding reasoning to the case when near atoms have time intervals.

Lemma 14 *Let $G = \{g_1, g_2\}$ be a go theory such that $obj(g_1) = o_1$ and $obj(g_2) = o_2$. $G \models \mathsf{near}(o_1, o_2, d, t_1, t_2)$ iff:*

1. *$TCI(g_1)$ and $TCI(g_2)$ are defined and*
2. *$[t_1, t_2] \in TCI(g_1)$ and $[t_1, t_2] \in TCI(g_2)$ and*
3. *$\forall t \in [t_1, t_2]$ $\mathsf{dist}(P^-(g_1, t), P^-(g_2, t)) \leq d$ and*
4. *$\forall t \in [t_1, t_2]$ $\mathsf{dist}(P^-(g_1, t), P^+(g_2, t)) \leq d$ and*
5. *$\forall t \in [t_1, t_2]$ $\mathsf{dist}(P^+(g_1, t), P^-(g_2, t)) \leq d$ and*
6. *$\forall t \in [t_1, t_2]$ $\mathsf{dist}(P^+(g_1, t), P^+(g_2, t)) \leq d$.*

The computation of items (3)–(6) of the above Lemma 14 is not straightforward because we are dealing with continuous time. If we can come up with simple equations that represent the change of $P^-(g,t)$ and $P^+(g,t)$ change w.r.t. time then we can effectively check the the items three to six of Lemma 14. The following lemma shows that for a given go atom g, $P^-(g,t)$ and $P^+(g,t)$ are piecewise linear functions.

Lemma 15 *Let $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a normalized go atom such that $TCI(g)$ is defined. If t is a time point in $TCI(g)$, then*

$$P^-(g,t) = \begin{cases} P_1 + v^-(t - t_1^+)\vec{g} & \text{if } t < t^* \\ P_2 + v^+(t - t_2^+)\vec{g} & \text{if } t^* \leq t \end{cases}$$

$$P^+(g,t) = \begin{cases} P_1 + v^+(t - t_1^-)\vec{g} & \text{if } t < T^* \\ P_2 + v^-(t - t_2^-)\vec{g} & \text{if } T^* \leq t \end{cases}$$

where $t^* \in \mathbf{R}$ such that $P_1 + v^-(t^* - t_1^+)\vec{g} = P_2 + v^+(t^* - t_2^+)\vec{g}$ and $T^* \in \mathbf{R}$ such that $P_1 + v^+(T^* - t_1^-)\vec{g} = P_2 + v^-(T^* - t_2^-)\vec{g}$.

The following example illustrates the use of this lemma.

Example 20 *Let*

$$g = go(o_1, (100, 100, 0), (100, 700, 0), 5, 50, 130, 180, 4, 6);$$

$$g' = go(o_2, (100, 500, 0), (100, 1300, 0), 50, 70, 180, 200, 5, 8).$$

Let $b = \mathsf{near}(o_1, o_2, 580, 80, 120)$. Figure 6.2 shows the positional certainty region for o_1 and o_2 for every time point in their temporal certainty interval. In addition, the piecewise linear functions $P^-(g,t), P^+(g,t), P^-(g',t)$ and $P^+(g',t)$ in the interval $[80, 120]$ are also displayed on the figure. An inspection of the figure allows us to conclude that the two objects are maximally far apart when o_1 is on $P^-(g,t)$ and o_2 is on $P^+(g',t)$. Furthermore there are times when the distance is greater than 580 (e.g. at time 120). Hence

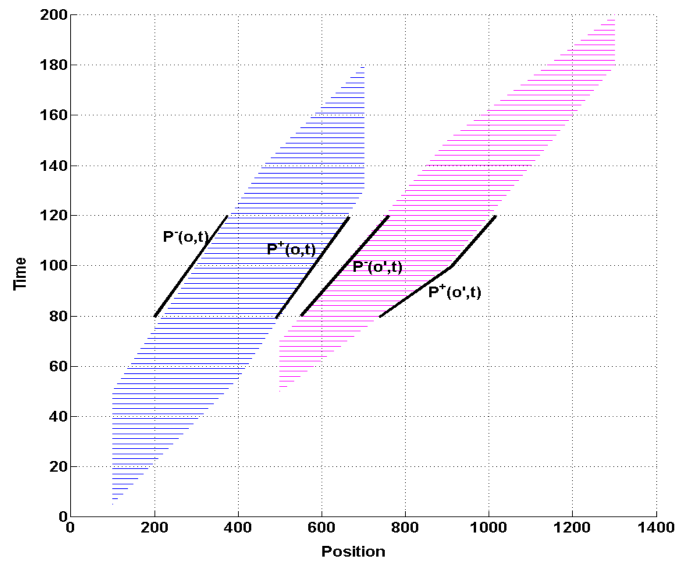


Figure 6.2: Graphical representation of positional certainty regions in Example 20

b is not a logical consequence of the go*-theory $G = \{g, g'\}$. We could however say that* $\mathsf{near}(o_1, o_2, 700, 80, 120)$ *is a logical consequence of G.*

6.3.3 Arbitrary go theories and arbitrary ground near atom.

We now extend this intuition to arbitrary *go*-theories and ground near atoms of the form $\mathsf{near}(o, o', d, t_1, t_2)$. For an arbitrary *go*-theory G, we need to consider all pairs of plans for G^o and $G^{o'}$ because they determine the positional and temporal intervals for the two objects. Given an object o, for every plan π, we need to consider two problems:

1. Can we predict the possible locations of o not only in the temporal certainty interval of a movement in π, but also during a time interval that spans several movements?
2. How can we represent the change in maximal and minimal advancement points during the interval $[t_1, t_2]$?

The short answer to the first question is "yes" – in some cases, we can predict the possible locations of o not only in the temporal certainty interval of a movement in π, but also during a time interval that spans over several several movements. The following lemma gives necessary and sufficient conditions when this can be done.

Lemma 16 *Let G be a go-theory, o be an object, and $\pi = \langle \Gamma, \sqsubseteq \rangle$ be a plan for G^o. Suppose $\gamma_1 \sqsubseteq \gamma_2 \ldots \sqsubseteq \gamma_n$ are the movements in π with $TCI(G^o, \pi, \gamma_k) = [T_k^-, T_k^+]$. $TCI(G^o, \pi, \gamma_i) \cup TCI(G^o, \pi, \gamma_{i+1}) \ldots \cup TCI(G^o, \pi, \gamma_j)$ is a* single time interval *iff for every $1 \leq i \leq k < j \leq n$ the following are true*

- $T_k^+ = T_{k+1}^-$
- $loc_2(\gamma_k) = loc_1(\gamma_{k+1})$

The following definition specifies when a set of movements is temporally relevant to a given time interval.

Definition 30 *Let G, o, π be as in Lemma 16. Suppose S is a subset of the set of all movements in π which satisfy the conditions in lemma 16. Then S is a **series of continuous movements** w.r.t. π. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the elements of S. Furthermore S is **temporally relevant** to time interval $[t_1, t_2]$ iff $t_1 \in TCI(G^o, \pi, \gamma_1)$ and $t_2 \in TCI(G^o, \pi, \gamma_n)$*

We now generalize Lemma 15 to show that $P^-(G^o, \pi, \gamma, t)$ and $P^+(G^o, \pi, \gamma, t)$ are piecewise linear functions as well.

Lemma 17 *Let G be a go-theory, o be an object, and π be a plan for G^o. Suppose γ is a movement in π such that $TCI(G^o, \pi, \gamma)$ is defined and $CheckPoints(\gamma) = [p_1, p_2 \ldots p_n]$. Then for every time point t in $TCI(G^o, \pi, \gamma)$, $P^+(G^o, \pi, \gamma, t)$ satisfies the following:*

$$P^+(G^o,\pi,\gamma,t) = \begin{cases} \ell_1^-(t) & \textit{if } T_1^- \leq t \leq T_2^- \wedge t < t_1^* \\ \ell_1^+(t) & \textit{if } T_1^- \leq t \leq T_2^- \wedge t \geq t_1^* \\ \vdots & \\ \ell_i^-(t) & \textit{if } T_i^- \leq t \leq T_{i+1}^- \wedge t < t_i^* \\ \ell_i^+(t) & \textit{if } T_i^- \leq t \leq T_{i+1}^- \wedge t \geq t_i^* \\ \vdots & \\ \ell_{n-1}^-(t) & \textit{if } T_{n-1}^- \leq t \leq T_n^- \wedge t < t_{n-1}^* \\ \ell_{n-1}^+(t) & \textit{if } T_{n-1}^- \leq t \leq T_n^- \wedge t \geq t_{n-1}^* \end{cases}$$

where

- $T_i^- = T^-(G^o,\pi,\gamma,p_i)$
- $\ell_i^-(t) = p_i + \vec{\gamma}\left[(t - T_i^-) \times v^+(\gamma,p_i,p_{i+1})\right]$
- $\ell_i^+(t) = p_{i+1} + \vec{\gamma}\left[(t - T_{i+1}^-) \times v^-(\gamma,p_i,p_{i+1})\right]$
- $t_i^* \in \mathbf{R}$ *such that* $\ell_i^-(t_i^*) = \ell_i^+(t_i^*)$

Similarly $P^-(G^o,\pi,\gamma,t)$ *is piecewise linear.*

We are now ready to present necessary conditions for a go theory to entail a ground $\mathsf{near}()$-atom with respect to a specific plan for each object.

Theorem 9 *Let G be a go-theory, o, o' be objects, π, π' be plans for G^o and $G^{o'}$ respectively and $b = \mathsf{near}(o,o',d,t_1,t_2)$ be a ground atom. All instances of π and π' satisfy b iff all the following conditions hold:*

1. *there is a subset S of the set of movements in π such that S is temporally relevant to $[t_1, t_2]$*

2. *there is a subset S' of the set of movements in π' such that S' is temporally relevant to $[t_1, t_2]$*

3. $\forall t \in [t_1, t_2]\, \exists \gamma \in S \wedge \exists \gamma' \in S'$ *such that* $t \in TCI(G^o, \pi, \gamma)$ *and* $t \in TCI(G^{o'}, \pi', \gamma')$ *and*

   - $\mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)) \leq d$ *and*
   - $\mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)) \leq d$ *and*
   - $\mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)) \leq d$ *and*
   - $\mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)) \leq d$ *and*

The algorithm CheckNear (see Algorithm 6.8) checks if a ground near atom $\mathsf{near}(o, o', d, t_1, t_2)$ is entailed by G w.r.t. plans pi, π'.

Theorem 10 *Suppose G is a consistent* go-*theory and $b = \mathsf{near}(o, o', d, t_1, t_2)$ is a ground atom. b is a logical consequence of G iff for every plan π for G^o, and π' of $G^{o'}$ algorithm* $\mathsf{CheckNear}(G, \pi, \pi', b)$ *returns "true".*

The above theorem says that to check if $G \models \mathsf{near}(o, o', d, t_1, t_2)$, we need to execute the CheckNear algorithm on all pairs of plans π, π' for these two objects.

6.4 Ground Atomic far Queries

Finding an algorithm to solve the far-entailment problem is a complex task. We are going to explain the algorithm using three cases in increasing complexity. We first consider the

Algorithm $\mathsf{CheckNear}(G, \pi, \pi', b))$

1. Suppose $b = \mathsf{near}(o, o', d, t_1, t_2)$
2. **if** $\mathcal{C}(pi)$ or $\mathcal{C}(pi')$ have no solution **then return** $true$
3. S= subset of set of movements in π that is temporally relevant to $[t_1, t_2]$
4. S'= subset of set of movements in π' that is temporally relevant to $[t_1, t_2]$
5. **if** S or S' not exists **then return** $false$
6. Let $T = \{t \mid \text{is an end point of } TCI(G^o, \pi, \gamma) \wedge \gamma \in S \wedge t_1 \leq t \leq t_2\}$
7. Let $T' = \{t \mid \text{is an end point of } TCI(G^{o'}, \pi', \gamma') \wedge \gamma' \in S' \wedge t \in [t_1, t_2]\}$
8. Let $T_1 \leq T_2 \leq \cdots \leq T_n$ be the elements of $T \cup T' \cup \{t_1, t_2\}$
9. **for** i **from** 1 **to** $n - 1$ **do**
   - Let $\gamma \in S$ such that $[T_i, T_{i+1}] \in TCI(G^o, \Gamma, \sqsubseteq, \gamma)$
   - Let $\gamma' \in S'$ such that $[T_i, T_{i+1}] \in TCI(G^{o'}, \Gamma', \sqsubseteq', \gamma')$
   - **if** $\exists t \mid \mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)) > d$ and $T_i \leq t \leq T_{i+1}$ **then return** $false$
   - **if** $\exists t \mid \mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)) > d$ and $T_i \leq t \leq T_{i+1}$ **then return** $false$
   - **if** $\exists t \mid \mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)) > d$ and $T_i \leq t \leq T_{i+1}$ **then return** $false$
   - **if** $\exists t \mid \mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)) > d$ and $T_i \leq t \leq T_{i+1}$ **then return** $false$
10. **return** $true$

Algorithm 6.8: CheckNear algorithm

case when there are only two atoms per object and the far atom defines a single time point instead of an interval. We then explain how to generalize the idea for arbitrary far atoms with intervals. Finally we present the case where the go theory may contain more than one atom per object and we have arbitrary far atoms.

6.4.1 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 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 18 *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 $PCR(g, t)$ and $PCR(g', t)$ is greater than d.*

Note that the minimum distance between two line segments can be computed in constant time [51].

Example 21 *Let $g = \mathsf{go}(o, (40, 10, 0), (70, 50, 0), 12, 13, 21, 21, 4, 10)$ and $g' = \mathsf{go}(o', (55, 20, 0), (45, 80, 0), 17, 18, 32, 33, 2, 6)$ 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 time point 19. At time 19, o is somewhere on the line segment $PCR(g, 19) = [(58, 34, 0), (65.2, 43.6, 0)]$ and object o' is on the line segment $PCR(g', 19) = [(54.67, 21.97, 0), (53.08, 31.84, 0)]$. 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 18 becomes: "*The **maximum** distance between line segments $PCR(g, t)$ and $PCR(g', t)$ is **less than or equal to** d*". This maximum distance is achieved at the end points of $PCR(g, t)$ and $PCR(g', t)$, e.g. $P^-(g, t)$ and $P^+(g', t)$. However the minimum distance between $PCR(g,t)$ and $PCR(g',t)$ is not necessarily at the end points hence its computation is more complex.

6.4.2 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 $PCR(g,t)$ and as shown in section 6.3 end points of $PCR(g,t)$ 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 Section 6.3.

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,z,t)-quadruples such that there exists a model $\mathcal{I}$ of g in which $\mathcal{I}(obj(g),t) = (x,y,z)$. In other words, it defines where and when it is *possible* for object o to be.

Definition 31 *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 $\{(x,y,z,t) \mid t \in T$ and $(x,y,z) \in PCR(g,t)\}$.*

Lemma 19 *$SE(g,T)$ is a convex set.*

Example 22 *Let $g' = \mathsf{go}(o', (55,20,0), (45,80,0), 17, 18, 32, 33, 2, 6)$ be a go atom. The space envelope of g' over time interval $[19,21]$, is shown in Figure 6.3. It is easy to see that $SE(g',[19,21])$ is convex.*

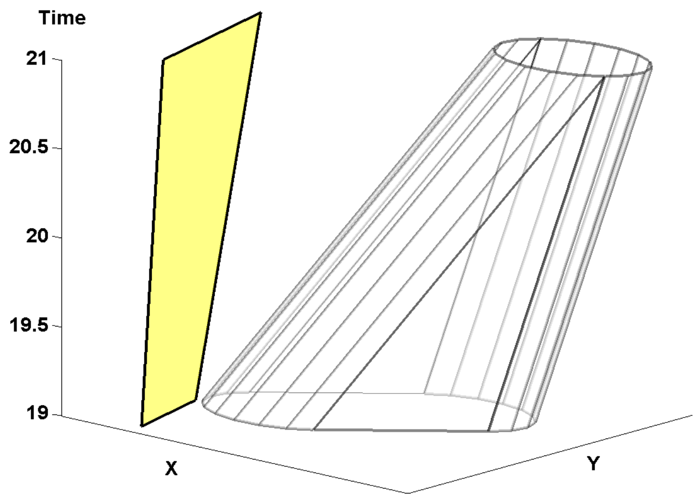


Figure 6.3: The polygon on the left is the space envelope of g' during $[19, 21]$ and the volume on the right is 5-neighborhood of g during time interval $[19, 21]$. g and g' are as defined in Examples 22,23 and 24.

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 32 (d-neighborhood) *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-neighborhood** of g during T, denoted $Nbr(g, T, d) = \{(x, y, z, t) \mid t_1 \leq t \leq t_2$ and $(x, y, z) \in NearPts(g, t, d)\}$ where $NearPts(g, t, d) = \{(x, y, z) \mid \exists(x', y', z') \in PCR(g, t)$ and $\mathsf{dist}((x, y, z), (x'y', z')) \leq d\}$.*

Intuitively $NearPts(g, t, d)$ is the set of all points p such that all points on the line segment $PCR(g, t)$ 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, z, t) such that it is possible for $obj(g)$ to be within d units of (x, y, z) at some time t in interval T.

Lemma 20 (i) *$Nbr(g, T, d)$ is a convex set.* (ii) *If $d = 0$, then $Nbr(g, T, d) = SE(g, T)$.*

Example 23 *Let $g = \mathsf{go}(o, (40, 10, 0), (70, 50, 0), 12, 13, 19, 21, 4, 10)$ be a go atom. The 5-neighborhood of g over time interval $[19, 21]$ is shown on the far right hand side of Figure 6.3. It is easy to see that it is a convex set.*

The following lemma states the necessary and sufficient conditions under which the binary theory models a ground far atom (i.e. $\{g, g'\} \models f$).

Lemma 21 *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. [52] provides polynomial algorithms to check for the intersection of two convex sets — these can be used directly to check the second condition above.

Example 24 *Let $g = \mathsf{go}(o, (40, 10, 0), (70, 50, 0), 12, 13, 21, 21, 4, 10)$ and $g' = \mathsf{go}(o', (55, 20, 0), (45, 80, 0), 17, 18, 32, 33, 2, 6)$ 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 6.3 shows $Nbr(g, [19, 21], 5)$ and $SE(g', [19, 21])$. It is apparent from the figure that the two do not intersect: hence $G \models f$.*

6.4.3 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 plans of G^o. We now generalize definitions 31 and 32 to accommodate non-binary go-theories with possibly multiple plans.

Definition 33 ($SE(G^o, \pi, T)$) *Let G be a go-theory, o be an object, and π be a plan for G^o. If $T = [t_1, t_2]$ is a time interval there is a subset S of movements in π that is temporally relevant to T , then $SE(G^o, \pi, T)$ is the set of all points (x, y, z, t) such that*

- *$t \in T$ and $t \in TCI(G^o, \pi, \gamma)$ for some $\gamma \in S$,*
- *(x, y, z) is on $PCR(G^o, \pi, \gamma, t)$.*

$SE(G^o, \pi, T)$ is not defined if there is no series of continuous movements S w.r.t. π, that is temporally relevant to T.

Note that $SE(G^o, \pi, T)$ is not necessarily convex when T spans over multiple movements.

We can generalize Nbr to $Nbr(G^o, \pi, T, d)$ in a similar manner.

Definition 34 ($Nbr(G^o, \pi, T, d)$) *Let G be a go-theory, o be an object, and π be a plan for G^o. Suppose $T = [t_1, t_2]$ is a time interval and there is a subset S of movements in π that is temporally relevant to T. Then given a real number d,*

$$Nbr(G^o, \pi, T, d) = \{(x, y, z, t) \mid t \in T \textit{ and } (x, y, z) \in NearPts(G^o, \pi, t, d)\}$$

where $NearPts(G^o, \pi, t, d) = \{(x, y, z) \mid \exists (x', y', z') \in PCR(G^o, \pi, \gamma, t)$ for some $\gamma \in S$ such that $t \in TCI(G^o, \pi, \gamma)$ and $\mathsf{dist}((x, y, z), (x'y', z')) \leq d\}$.

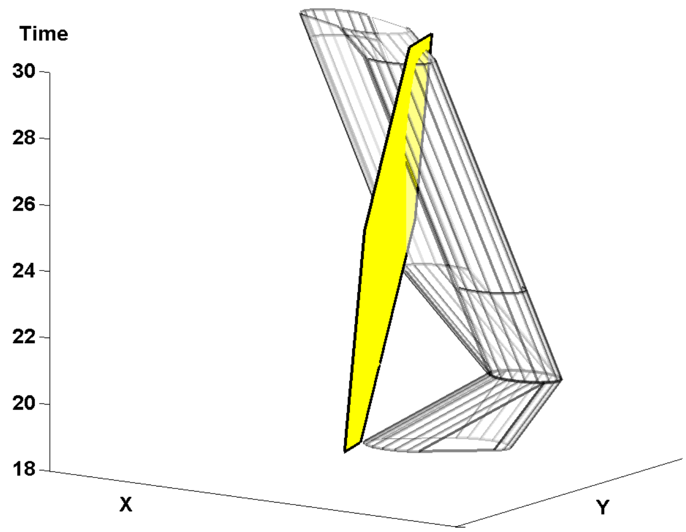


Figure 6.4: The space envelope of o' with respect to G and 5-neighborhood of o w.r.t. G, π and π' during time interval $[19, 30]$. G, π and π' are as defined in Example 25

Just like the space envelope in arbitrary theories when T spans over multiple movements , $Nbr(G^o, \pi, T, d)$ is not necessarily convex.

We now state a theorem describing the conditions under which a go theory G entails a $\mathsf{far}()$ atom.

Theorem 11 *Suppose G is a go-theory and o, o' are objects.Let π and π' be plans of G^o and $G^{o'}$. The ground atom $f = \mathsf{far}(o, o', d, t_1, t_2)$ is satisfied in all instances of π and π' iff all the following hold:*

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

Example 25 *Let $f = \mathsf{far}(o, o', 5, 19, 30)$ and $G = \{g_1, g_2, g'\}$ such that*

Algorithm $\mathsf{CheckFar}(G, \pi, \pi', f)$

1. Suppose $f = \mathsf{far}(o, o', d, t_1, t_2)$
2. Let $S_o \subseteq \pi$ such that S_o is temporally relevant to $[t_1 t_2]$
3. Let $S_{o'} \subseteq \pi'$ such that $S_{o'}$ is temporally relevant to $[t_1 t_2]$
4. **if** no such S_o or $S_{o'}$ exists **then return** false
5. Let $T_1, T_2 \ldots T_n$ be the convex partition of $[t_1, t_2]$
6. **for each** $i \leq n$ **do**
   - **if** $Nbr(G^o, \pi, T_i, d) \cap SE(G^{o'}, \pi', T_i]) \neq \emptyset$ **then return** false
7. **return** true

Algorithm 6.9: CheckFar algorithm

- $g_1 = \mathsf{go}(o, (40, 10, 0), (70, 50, 0), 12, 13, 21, 21, 4, 10)$,
- $g_2 = \mathsf{go}(o, (70, 50, 0), (30, 80, 0), 20, 21, 30, 31, 4, 10)$,
- $g' = \mathsf{go}(o', (55, 20, 0), (45, 80, 0), 17, 18, 32, 33, 2, 6)$.

If $g_1 \sqsubseteq g_2$ are the movements in π (plan for G^o) and π' (plan for $G^{o'}$) has the single movement $\{g'\}$ then Figure 6.4 shows $Nbr(G^o, \pi, [19, 30], 5)$ and $SE(G^{o'}, \pi', [19, 30])$. The figure also shows that the sets intersect - hence $G \not\models f$.

As $Nbr(G^o, \pi, T, d)$ and $SE(G^{o'}, \pi', T)$ are not always convex, computing their intersection is tricky. We may however 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 $T^+(G, \pi, \gamma, P)$ or $T^-(G, \pi, \gamma, P)$ for some $P = loc_1(g)$ or $P = loc_2(g)$ where $g \in G^o \cup G^{o'}$. It is easy to verify that $n \leq |G^o| + |G^{o'}|$. For each T_i, $Nbr(G^o, \pi, T_i, d)$ and $SE(G^{o'}, \pi', T_i)$ are convex and this is used in the CheckFar algorithm (see Algorithm 6.9).

Theorem 12 *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 plan π and π' of G^o and $G^{o'}$, the algorithm* CheckFar*(G, π, π', f) returns "true".*

Chapter 7

Answering Non-ground Atomic Queries

In this chapter, we present algorithms to answer *selected* non-ground queries w.r.t. *simple* go theories. We believe the queries we consider constitute the bulk of the interesting types of queries users will ask. Our algorithms return answer substitutions defined below.

7.1 Solution of a Non-ground Query

Definition 35 (Substitution) *A **substitution**[1] ϕ is a set of pairs (X, v) where X is a variable and v is a value such that if X ranges over $\mathbf{R}$ (resp. $\mathbf{OID}, \mathbf{P}$), then $v \in \mathbf{R}$ (resp. $\mathbf{OID}, \mathbf{P}$).*

***The application of substitution** ϕ **to an atom** A, denoted $\phi(A)$, is obtained by the replacement of all the occurrences of variable X in A by v iff $(X, v) \in \phi$.*

Definition 36 (Satisfaction of constraints by a substitution) *Let Θ be a set of constraints and $X_1, X_2 \ldots X_n$ be the variables in Θ. A **substitution** $\phi = \{(X_1, v_1), (X_2, v_2) \ldots (X_n, v_n)\}$ **satisfies** Θ iff all the constraints in Θ are satisfied when each X_i in Θ is replaced with v_i.*

Definition 37 (Solution to an atom) *Let G be a go theory and A be an atom containing variables. A set of constraints Θ is a* solution to A with respect to G *iff for every*

[1] In classical logic, a substitution can allow variables to be replaced by other variables - we do not permit this.

substitution ϕ that satisfies Θ, $\phi(A)$ is ground and $G \models \phi(A)$.

7.2 go Queries

Algorithm CheckGo applies only to ground go atoms - in this section, we show how to build upon it to answer some (but not all) types of non-ground go atoms. The following result on ground go-atoms states that when all the time and velocity intervals of the first go-atom are subsets of the corresponding intervals of the second, then entailment of the first go-atom by a go-theory implies entailment of the second.

Lemma 22 *Let G be a go theory, $q = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a normalized ground go atom, and $Q = \mathsf{go}(o, P_1, P_2, T_1^-, T_1^+, T_2^-, T_2^+, V^-, V^+)$ be a ground go atom such that:*

$$T_1{}^- \leq t_1{}^- \leq t_1{}^+ \leq T_1{}^+ \text{ and } T_2{}^- \leq t_2{}^- \leq t_2{}^+ \leq T_2{}^+ \text{ and } V^- \leq v^- \leq v^+ \leq V^+$$

If $G \models q$ then $G \models Q$.

The following definition associates a set of constraints with a go-atom, a go-theory, a plan and a movement of the plan.

Definition 38 (Constraint Set $\Theta_{q,G^o,\gamma}$) *Let $q = \mathsf{go}(o, p_1, p_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a go atom, G be a consistent simple go theory, and let π be the main plan for G^o. Suppose γ is a movement in π such that $LS(\gamma) = [P_1, P_2]$. The set of constraints $\Theta_{q,G^o,\gamma}$ contains the following*

- $0 \leq d_1 \leq d_2 \leq \mathsf{dist}(P_1, P_2)$, *where d_1, d_2 are variables in $V_{\mathbf{R}}$*

- $p_1 = P_1 + d_1 * \vec{\gamma}$
- $p_2 = P_1 + d_2 * \vec{\gamma}$
- $t_1^- \leq T^-(G^o, \pi, \gamma, p_1) \leq T^+(G^o, \pi, \gamma, p_1) \leq t_1^+$
- $t_2^- \leq T^-(G^o, \pi, \gamma, p_2) \leq T^+(G^o, \pi, \gamma, p_2) \leq t_2^+$
- $v^- \leq V^-(G^o, \pi, \gamma, p_1, p_2) \leq V^+(G^o, \pi, \gamma, p_1, p_2) \leq v^+$

The following theorem states that there is a one to one correspondence between the solutions of the above constraints and the substitutions that satisfy a given (ground or non-ground) query.

Theorem 13 *Let G be a consistent simple go theory and* $q = \mathsf{go}(o, p_1, p_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *be a go atom. Suppose* π *is the plan for* G^o*. There is a solution to* q *w.r.t.* G *iff there is a movement* γ *in* π *such that* $\Theta_{q,G^o,\gamma}$ *is satisfiable. Furthermore* $\Theta_{q,G^o,\gamma}$ *is a solution to* q *w.r.t.* G*.*

An immediate consequence of this theorem is that given a go-theory G we can find a solution to a non-ground go atom q by: (i) first fixing an o, (ii) then finding a movement γ for π, and (iii) checking if $\Theta_{q,G^o,\gamma}$ is solvable. If so, the satisfying substitution is an answer to query q. Repeating this for different o's and γ's yields the entire set of solutions to q.

Though the number of satisfying assignments for each solution is infinite, there are at most n solutions to q w.r.t. G where n is the number of go atoms in G^o. When p_1 and/or p_2 are bounded in q, the number of solutions will significantly decrease since the only solvable $\Theta_{q,G^o,\gamma}$ are those for movements γ such that $[p_1, p_2] \in LS(\gamma)$.

We now consider the problem of checking if there is a substitution that satisfies $\Theta_{q,G^o,\gamma}$. $\Theta_{q,G^o,\gamma}$ is solvable when only the object term is ground in q. If this is not so, we

Algorithm Solve($G, \Theta_{q,G^o,\gamma}$)

1. **If** γ is not spatially related to q **then return** fail
2. $T_1^- = T^-(G^o, \pi, \gamma, p_1)$ and $T_1^+ = T^+(G^o, \pi, \gamma, p_1)$
3. **if not** $t_1^- \leq T_1^- \leq T_1^+ \leq t_1^+$ **then return** fail
4. $T_2^- = T^-(G^o, \pi, \gamma, p_2)$ and $T_2^+ = T^+(G^o, \pi, \gamma, p_2)$
5. $V^- = V^-(G^o, \pi, \gamma, p_1, p_2)$ and $V^+ = V^+(G^o, \pi, \gamma, p_1, p_2)$
6. $\Theta^*_{q,G^o,\gamma}$ has the following constraints:
   - $?t_2^- \leq T_2^- \leq T_2^+ \leq ?t_2^+$
   - $?v^- \leq V^- \leq V^+ \leq ?v^+$
7. **return** $\Theta^*_{q,G^o,\gamma}$

Algorithm 7.10: Algorithm for go-queries with fixed locations.

can easily check satisfiability of $\Theta_{q,G^o,\gamma}$ because $T^-(G^o, \pi, \gamma, p_1)$ and $T^+(G^o, \pi, \gamma, p_1)$ are piecewise linear functions [2] and $V^-(G^o, \pi, \gamma, p_1, p_2)$ and $V^+(G^o, \pi, \gamma, p_1, p_2)$ are piecewise constant functions.

We now present some example algorithms that check if $\Theta_{q,G^o,\gamma}$ is solvable when some terms in the query are ground.

Go-queries with fixed locations These queries have the form $q = go(o, p_1, p_2, t_1^-, t_1^+, ?t_2^-, ?t_2^+, ?v^-, ?v^+)$ where any $?x$ is a variable. Let G be a simple go theory, π be the main plan for G^o andγ be a movement in π. The Algorithm 7.10 can be used to check if $\Theta_{q,G^o,\gamma}$ has a consistent substitution. Note that if the algorithm does not fail at steps 1 or 3 then the first four constraints in $\Theta_{q,G^o,\gamma}$ are always satisfied. The algorithm returns the tightened constraint set $\Theta_{q,G^o,\gamma}$ subject to the ground variables which are always satisfiable.

[2] As shown in Section 7.3 the maximum/minimum advancement points which are inverse functions of $T^-(G^o, \pi, \gamma, p_1)$ and $T^+(G^o, \pi, \gamma, p_1)$ are piecewise linear functions.

Algorithm Solve$(G, \Theta_{q,G^o,\gamma})$

1. **if** γ is not temporally related to q **then return** fail
2. **if** $T^-(G^o, \pi, \gamma, loc_1(\gamma)) \geq t_1^-$ **then** $p^- = loc_1(\gamma)$
   else find p^- such that $T^-(G^o, \pi, \gamma, p^-) = t_1^-$
3. **if** $T^+(G^o, \pi, \gamma, loc_2(\gamma)) \leq t_1^+$ **then** $p^+ = loc_2(\gamma)$
   else find p^+ such that $T^+(G^o, \pi, \gamma, p^+) = t_1^+$
4. **if** p^- is after p^+ **then return** fail
5. **if** $T^-(G^o, \pi, \gamma, loc_1(\gamma)) \geq t_2^-$ **then** $q^- = loc_1(\gamma)$
   else find q^- such that $T^-(G^o, \pi, \gamma, q^-) = t_2^-$
6. **if** $T^+(G^o, \pi, \gamma, loc_2(\gamma)) \leq t_2^+$ **then** $q^+ = loc_2(\gamma)$
   else find q^+ such that $T^+(G^o, \pi, \gamma, q^+) = t_2^+$
7. **if** q^- is after q^+ **then return** fail
8. $\Theta^*_{q,G^o,\gamma}$ has the following constraints:
   - $\mathsf{dist}(loc_1(\gamma), p^-) \leq d_{1,\pi} \leq d_{2,\pi} \leq \mathsf{dist}(loc_1(\gamma), q^+)$
   - $p_1 = P_1 + d_{1,\pi} * \vec{\gamma}$
   - $p_2 = P_1 + d_{2,\pi} * \vec{\gamma}$
   - $v^- \leq V^-(G^o, \pi, \gamma, p_1, p_2) \leq V^+(G^o, \pi, \gamma, p_1, p_2) \leq v^+$
9. **return** $\Theta^*_{q,G^o,\gamma}$

Algorithm 7.11: Algorithm for go-queries with fixed time.

Go-queries with fixed time These queries have the form $q = go(o, ?p_1, ?p_2, t_1^-, t_1^+, t_2^-, t_2^+, ?v^-, ?v^+)$ where any $?x$ is a variable. Let G be a simple go theory, π be the main plan for G^o, γ be a movement in π. The Algorithm 7.11 can be used to check if $\Theta_{q,G^o,gamma}$ has a consistent substitution. The algorithm also tightens the constraint set $\Theta_{q,G^o,\gamma}$ subject to the ground variables.

7.3 in Queries

In this section, we develop algorithms to find answer substitutions to $\mathsf{in}()$ queries w.r.t. *simple* go-theory. We will initially assume that the object in an $\mathsf{in}()$-atom is ground. Subsequently, we will show how to relax this assumption.

The following lemmas are not hard to prove.

Lemma 23 *Let G be a go theory, $q = \mathsf{in}(o, p_1, p_2, t_1, t_2)$ and $Q = \mathsf{in}(o, P_1, P_2, t_1, t_2)$ be ground atoms such that:*

- *${P_1}^x \leq {p_1}^x$ and ${P_1}^y \leq {p_1}^y$ and ${P_1}^z \leq {p_1}^z$.*
- *${P_2}^x \geq {p_2}^x$ and ${P_2}^y \geq {p_2}^y$ and ${P_2}^z \geq {p_2}^z$.*

If $G \models q$ then $G \models Q$.

Lemma 24 *Let G be a go theory, $q = \mathsf{in}(o, p_1, p_2, t_1, t_2)$ and $Q = \mathsf{in}(o, p_1, p_2, T_1, T_2)$ be ground atoms such that $T_1 \leq t_1$ and $T_2 \geq t_2$. If $G \models q$ then $G \models Q$.*

As the lemmas suggest, there are infinitely many ground $\mathsf{in}()$ atoms that can be a logical consequence of a go theory. We represent those solutions finitely by defining minimal solutions. To do this, we now define **minimal volumes** and **minimal time windows**.

Definition 39 (Minimal Volume) *Let G be a go theory and $q = \mathsf{in}(o, p_1, p_2, t_1, t_2)$ be a ground atom such that $G \models q$. The volume $Vol(q)$, is a* **minimal volume** *for time interval $[t_1, t_2]$ if there does not exist a ground atom $Q = \mathsf{in}(o, P_1, P_2, t_1, t_2)$ such that $G \models Q$ and*

- *${p_1}^x \leq {P_1}^x$ and ${p_1}^y \leq {P_1}^y$ and ${p_1}^z \leq {P_1}^z$ and*
- *${p_2}^x \geq {P_2}^x$ and ${p_2}^y \geq {P_2}^y$ and ${p_2}^z \geq {P_2}^z$.*

Definition 40 (Minimal Time Window) *Let G be a go theory and $q = \mathsf{in}(o, p_1, p_2, t_1, t_2)$ be a ground atom such that $G \models q$. The time interval $[t_1, t_2]$ is a* **minimal time window** *for volume $Vol(q)$, if there does not exist a ground atom $Q = \mathsf{in}(o, p_1, p_2, T_1, T_2)$ such that $G \models Q$ and $[T_1, T_2] \subset [t_1, t_2]$.*

For a consistent simple go theory, we know that all plans are equal to the main plan π. In this case, any volume that encloses a subsegment of $LS(\gamma)$ where γ is a movement in π, and any time window that contains the time interval in which the object is traveling on L, will be a logical consequence of G. The following definition associates a set of constraints with a given $\mathsf{in}()$-query, an object, a plan, and a movement in the plan.

Definition 41 *Let G be a simple consistent go theory and $q = \mathsf{in}(o, P, Q, t_1, t_2)$ be an atom. Suppose π is the main plan for G^o and suppose γ is a movement in π such that $LS(\gamma) = [P_1, P_2]$.We define the set $\Omega_{q,G^o,\gamma}$ of constraints to consist of the following:*

- $0 \leq d_1 \leq d_2 \leq \mathsf{dist}(P_1, P_2)$, *where* $d_1, d_2 \in \mathbf{R}$
- $p_1 = P_1 + d_1 * \vec{\gamma}$ *and* $p_2 = P_1 + d_2 * \vec{\gamma}$
- $t_1 \leq T^-(G^o, \pi, \gamma, p_2)$ *and* $T^+(G^o, \pi, \gamma, p_1) \leq t_2$ *and* $t_1 \leq t_2$,
- $P^x \leq min(p_1^x, p_2^x)$, $P^y \leq min(p_1^y, p_2^y)$, $P^z \leq min(p_1^z, p_2^z)$,
- $Q^x \geq max(p_1^x, p_2^x)$, $Q^y \geq max(p_1^y, p_2^y)$, $Q^z \geq max(p_1^z, p_2^z)$,

The following theorem tells us that to find a solution, Ω, to a $\mathsf{in}()$ atom q, we merely need to find a movement w.r.t. q which has a solvable set of associated constraints.

Theorem 14 *Let G be consistent simple go theory and let $q = \mathsf{in}(o, P, Q, t_1, t_2)$ be an atom. Suppose π is the main plan for G^o. There is a solution to q w.r.t. G iff there is a movement γ in π such that there exists a substitution that satisfies the constraints in $\Omega_{q,G^o,\gamma}$. Furthermore $\Omega_{q,G^o,\gamma}$ is a solution to q w.r.t. G.*

Notice that although the set of substitutions for each solution Ω is infinite, there will be at most n solutions where n is the number of go atoms in G^o. Checking if $\Omega_{q,G^o,\gamma}$ has

Algorithm Solve$(G, \Omega_{q,G^o,\gamma})$

1. **if** $t_1 > T^-(G^o, \pi, \gamma, loc_2(\gamma))$ or $t_2 < T^+(G^o, \pi, \gamma, loc_1(\gamma))$
   then return fail
2. **if** $T^-(G^o, \pi, \gamma, loc_1(\gamma)) \geq t_1$ **then** $q = loc_1(\gamma)$
   else find q such that $T^-(G^o, \pi, \gamma, q) = t_1$
3. **if** $T^+(G^o, \pi, \gamma, loc_2(\gamma)) \leq t_2$ **then** $p = loc_2(\gamma)$
   else find p such that $T^+(G^o, \pi, \gamma, p) = t_2$
4. **if** p is after q **then** $\Omega^*_{q,G^o,\gamma}$ has the following constraints:
   - $\mathsf{dist}(loc_1(\gamma), q) \leq d \leq \mathsf{dist}(loc_1(\gamma), p)$
   - $p_1 = loc_1(\gamma)) + d * \vec{\gamma}$
   - $?P^x \leq p_1^x \leq ?Q^x$, $?P^y \leq p_1^y \leq Q^y$, $?P^z \leq p_1 \leq Q^y$,

   else $\Omega^*_{q,G^o,\gamma}$ has the following constraints:
   - $?P^x \leq min(p^x, q^x)$, $?P^y \leq min(p^y, q^y)$, $?P^z \leq min(p^z, q^z)$,
   - $?Q^x \geq max(p^x, q^x)$, $?Q^y \geq max(p^y, q^y)$, $?Q^z \geq max(p^z, q^z)$,
5. **return** $\Omega^*_{q,G^o,\gamma}$

Algorithm 7.12: Algorithm for in-queries with fixed region.

a solution is easy given that all constraints are linear. The following algorithms illustrate how to do this for a partially instantiated query templates:

In-queries with fixed region These queries have the form $q = \mathsf{in}(o, ?P, ?Q, t_1, t_2)$ where any $?x$ is a variable. Let G be a simple go theory, π be the main plan for G, and let γ be a movement in π. The Algorithm 7.12 can be used to check if $\Omega_{q,G^o,\gamma}$ has a consistent substitution. The algorithm also tightens the constraint set $\Omega_{q,G^o,\gamma}$ subject to the ground variables.

In-queries with fixed time: These queries are of the form $q = \mathsf{in}(o, P, Q, ?t_1, ?t_2)$ where any $?x$ is a variable. Let G be a simple go theory, π be the main plan for G, and γ be a movement in π. The Algorithm 7.13 can be used to check if $\Omega_{q,G^o,\gamma}$ has a consistent

Algorithm Solve$(G, \Omega_{q,G^o,\gamma})$

1. **if** γ is not spatially related to q **then return** fail
2. Let $[R_1, R_2] = LS(\gamma) \cap Vol(q)$ such that $\mathsf{dist}(loc_1(\gamma)), R_1) \leq \mathsf{dist}(loc_1(\gamma)), R_2)$
3. $\Omega^*_{q,G^o,\gamma}$ has the following constraints:
   - $?t_1 \leq ?t_2$
   - $?t_1 \leq T^-(G^o, \pi, \gamma, R_2)$
   - $T^+(G^o, \pi, \gamma, R_1) \leq ?t_2$,
4. **return** $\Omega^*_{q,G^o,\gamma}$

Algorithm 7.13: In-queries with fixed time

substitution. Note that as long as the movement is spatially relevant to q, $\Omega_{q,G^o,\gamma}$ has a solution. The algorithm also tightens the constraint set $\Omega_{q,G^o,\gamma}$ subject to the ground variables.

Let us now revisit the case where the object o in an atom $a = \mathsf{in}(o, P, Q, t_1, t_2)$ is a variable. A straightforward solution is to use the approach described in this section by instantiating o to all possible objects in the go-theory. If for any instantiation, o_i, there is a solution, then $o = o_i$ will be inserted into the solution. Notice that this simple approach will produce an algorithm that is linear in the number of objects in the go theory. We can improve this result by pruning objects that are not relevant for a.

Using Indexes. We can define an index for the case when o is a variable and some of P, Q, t_1 and t_2 are ground. The *Temporal envelope* of G_o is a time interval $[T_1, T_2]$ where

- $T_1 = min\{t_1^-(g) \mid g \in G_o\}$.
- $T_2 = max\{t_2^+(g) \mid g \in G_o\}$.

The *bounding box* for G_o is a minimal volume that contains the set $\{[loc_1(g), loc_2(g)] | \ g \in G_o\}$ of line segments.

Consider the query $a = \mathsf{in}(o, P, Q, t_1, t_2)$ where o is a variable and all other parameters are known. Clearly, we only need to consider the objects satisfying two conditions: (i) the object should have a bounding box which intersects the query volume $[P, Q]$ and (ii) the object's temporal envelope must intersect the query interval $[t_1, t_2]$. Hence we will not do any computation for objects that are not relevant to the query. Using bounding boxes and temporal envelopes as indexes can be useful even when only some of P, Q, t_1 and t_2 are known. For example if P and Q are known, the bounding box will be effective in pruning.

7.4 near Queries

In order to answer near queries with variables, we can generalize the approach we used when answering ground queries. Once again, we assume initially that the object terms o and o' are ground. If we can express the maximum distance between any two object as a function of time, then we can express constraints with respect to the distance and the time interval specified in the query.

Definition 42 (Max distance between objects) *Let o and o' be two objects in a consistent* go-*theory G, and let t be a real number representing a point in time. The* maximum distance between o and o' with respect to G *is:*

$$\Delta_G(o, o', t) = max\{\mathsf{dist}(I(o, t), I(o't)) \mid I \models G\}.$$

We now address the problem of computing $\Delta_G(o, o', t)$?. When G is a simple go theory, we know that there is a main plan π for G^o and a main plan π' for $G^{o'}$. Suppose γ is a movement of π and γ' is a movement of π'. Then we can compute $\Delta_G(o, o', t)$ in the following manner:

- For every t in $TCI(G^o, \pi, \gamma)$ and $TCI(G^{o'}, \pi', \gamma')$

$$\begin{aligned}\Delta_G(o, o', t) = max(\; & \mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)), \\ & \mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)), \\ & \mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)), \\ & \mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)))\end{aligned}$$

- ∞, otherwise.

It is easy to show that $\Delta_G(o, o', t)$ is a piecewise quadratic function. This is due to the fact that each $P^-(G^o, \pi, \gamma, t)$ and $P^-(G^o, \pi, \gamma, t)$ is piecewise linear. We now define the critical time points for $\Delta_G(o, o', t)$.

Lemma 25 *Let G be a simple go theory and let o, o' be two objects. Then there are disjoint time intervals $T_1, T_2, \ldots, T_n$ and quadratic functions $f_1(t), f_2(t), \ldots f_n(t)$ such that $\Delta_G(o, o', t) = d$ iff $f_i(t) = d$ for some i such that $t \in T_i$.*

The following definition specifies critical time points that need to be considered when answering near queries.

Definition 43 (Critical time points for $\Delta_G(o, o', t)$) *Let G be a simple go theory and o, o' be two objects. Suppose time intervals $T_1, T_2, \ldots, T_n$ and quadratic functions $f_1(t)$, $f_2(t), \ldots f_n(t)$ satisfy the condition in Lemma 25. Then each end point of T_i is a **critical time point** of $\Delta_G(o, o', t)$.*

The following lemma states that the total number of critical time points of $\Delta_G(o, o', t)$ is linear with respect to the total number of go atoms in G^o and $G^{o'}$ when G is a simple theory.

Lemma 26 *Let G be a simple go theory and o, o' be two objects. The total number of critical time points in $\Delta_G(o, o', t)$ is bounded by $O(n)$ where n is the total number of go atoms in G^o and $G^{o'}$.*

As in the case of in queries, near-queries can have infinitely many answers. Our notion of a solution to a near query needs to capture all these answers. The following lemma demonstrates two characteristics of ground near queries which lead to an infinite number of answers for the non-ground case.

Lemma 27 *Let G be a* go *theory and $q = near(o, o', D, T_1, T_2)$ be a ground near atom such that $G \models q$. Then the following are true:*

- *$G \models near(o, o', d, T_1, T_2)$ where $D \leq d$.*
- *$G \models near(o, o', D, t_1, t_2)$ where $T_1 \leq t_1 \leq t_2 \leq T_2$.*

The following result defines a set of constraints whose solutions precisely capture the answers to a near-query.

Theorem 15 *Let G be consistent, simple go theory and $q = near(o, o', d, t_1, t_2)$ be a near atom. Suppose $\Psi_{q,G}$ is a constraint set containing the following constraints:*

- $T_1 \leq t_1 \leq t_2 \leq T_2$
- $D \leq d$

Algorithm Solve($G, \Psi_{q,G}$)

1. Let π be the plan for G^o and π' be the plan for $G^{o'}$
2. **if** there is no set of movements in π that is temporally relevant to $[t_1, t_2]$ or there is not a set of movements in π' that is temporally relevant to $[t_1, t_2]$ **then return** fail
3. Let T be the set of be the critical points of $\Delta_G(o, o', t)$
4. $D = max\{\Delta_G(o, o', t) \mid t \in \{t_1, t_2\} \cup T$ and $t_1 \leq t \leq t_2\}$
5. **return** the constraint $D \leq ?d$

Algorithm 7.14: Algorithm for near-queries with fixed time

- $\forall t \in [T_1, T_2], \Delta_G(o, o', t) \leq D$.

$\Psi_{q,G}$ is a solution to q w.r.t. G iff there is a substitution that satisfies $\Psi_{q,G}$.

Checking if $\Psi_{q,G}$ has a satisfying substitution can be efficiently done as $\Delta_G(o, o', t)$ is a piecewise quadratic function with O(n) critical time points where the function changes behavior.

We now illustrates how to do this for certain partially instantiated near queries:

Near-queries with fixed time: These queries are of the form $q = near(o, o', ?d, t_1, t_2)$ where any $?x$ is a variable. The Algorithm 7.14 can be used to check if $\Psi_{q,G}$ has a satisfying substitution when G is a simple go theory. The algorithm finds the maximum distance, D, between the two objects during the given time interval $[t_1, t_2]$. Any substitution of $?d$ that is greater than or equal to D will be a solution for $\Psi_{q,G}$.

Near-queries with fixed distance: These queries are of the form $q = near(o, o', d, ?t_1, ?t_2)$ where any $?x$ is a variable. The Algorithm 7.15 can be used to check if $\Psi_{q,G}$ has a satisfying substitution when G is a simple go theory. The algorithm finds the time intervals during which the distance between the two objects is less than or equal to d.

Algorithm Solve$(G, \Psi_{q,G})$

1. Let S be an empty list
2. Let $T_1, \ldots T_n$ be the critical time points of $\Delta_G(o, o', t)$
3. For $T_1, \ldots T_{n-1}$ do
   - Let $f_i(t)$ be the quadratic function with domain $[T_i, T_{i+1}]$ and $f_i(t) = \Delta_G(o, o', t)$
   - Solve for the time intervals that satisfy the inequality-system: $\{f_i(t) <= d \wedge T_i \leq t \leq T_{i+1}\}$
   - Insert these solutions into S
4. Merge the intervals in S if they form a single interval
5. **return** S

Algorithm 7.15: Algorithm for near-queries with fixed time

Any substitution of $?t_1, ?t_2$ that is within one of the computed time intervals will be a solution for $\Psi_{q,G}$. Note that there can be more than one time interval and the following algorithm computes all such intervals.

Variable objects and indexing. Indexing can be useful for $near(o, o', d, t_1, t_2)$ queries with object variables. When o and o' are variables, a naive algorithm can iteratively bind o, o' to all possible objects and check solvability of the resulting constraints - this will yield an algorithm that is quadratic in the number of atoms.

We can do better. Using the same index (bounding box and temporal envelopes) as for in()-queries, we can reduce the number of pairs of objects to be considered via two steps. In the first step, we prune any object if it's temporal interval and the query's temporal interval $[t_1, t_2]$ do not intersect. In the second step, for all remaining objects, prune the pair (o, o') if the minimum distance between the bounding box of o and the bounding box of o' is greater than d.

Chapter 8

Motion Closed World Assumption

8.1 Problem Definition and Motivation

Using the syntax and semantics defined in Chapter 2, one can make statements of the form "Object o is expected to leave location P_1 at some time point in the interval $[t_1^-, t_1^+]$ and reach location P_2 at some time point in the interval $[t_2^-, t_2^+]$ traveling at a velocity between v_1 and v_2. Go theories can be used, for example, to make statements such as "Plane p22 is expected to take off from Paris at some time between 10 and 12 and land at Boston at some time between 18 and 23 traveling at a speed between 10 to 20." Figure 8.1 shows the spatial layout of one such go-theory (the go theory is written in text at the top).

Example 26 *The* Planes *go-theory of Figure 8.1 is consistent as the interpretations $\mathcal{I}_1, \mathcal{I}_2$ below both satisfy it.*

- $\mathcal{I}_1$ *: $p22$ leaves Paris at time 11, flies to Boston at a constant speed of 14.85 and arrives in Boston at 19. $p22$ waits in Boston until 32, then it departs for Paris with a constant speed of 13.2 arriving in Paris at 41. The other plane, $p34$ leaves London at time 25 and flies to Delhi at a constant speed of 18.52, arriving in Delhi at 38.*
- $\mathcal{I}_2$ *: $p22$ leaves Paris at time 10, flies to Boston at constant speed of 14.85 and reaches Boston at 18. It waits in Boston until 19, when it takes off for Detroit where it arrives at time 21. It immediately departs and reaches Boston at time 29. At time*

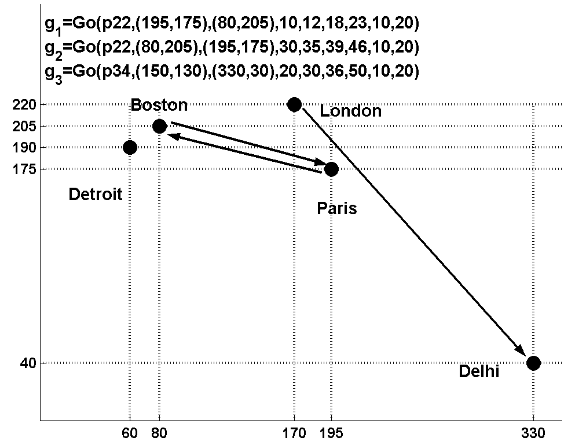


Figure 8.1: Planes example

30, $p22$ leaves Boston and flies to Paris at a constant speed of 11.88, arriving in Paris at time 40. The other plane, $p34$ leaves London at time 25 and flies to Delhi at a constant speed of 18.52, arriving in Delhi at 38.

It is important to note that even though $\mathcal{I}_2$ satisfies the Planes go theory, it is an interpretation that allows plane $p22$ to wander around in ways that were not explicitly stated in the Planes go theory. In particular, it lets the plane wander to Detroit which was never mentioned in the go-theory. In many applications it is important to exclude such "wandering" interpretations as they prevent us from making the intuitive (non-monotonic) inference that Plane $p22$ was never in Detroit.

The goal of this chapter is to ensure that intelligent negative inferences of this kind can be made from go-theories. In this chapter, a class of models of go-theories called *coherent* models are defined. Coherent models do not allow objects to move unless explicitly stated by the go theory. Next using this concept a *motion closed world assumption* (MCWA) and a notion of MCWA-entailment are defined. A major theoretical result is checking if a go-theory has a coherent model is NP-complete. Finally sound and complete algorithms to check if an in literal (positive or negative in atom) can be inferred from a go-theory using the MCWAare presented.

8.2 Coherence Definition

In this section, we define the concept of a coherent interpretation. We start by defining precedence of time intervals.

Definition 44 (Precedence) *Let $S = \{T_1, \ldots T_n\}$ be a set of time intervals, where $T_i = [t_{i1}, t_{i2}]$ for each i. T_i* ***immediately precedes*** *T_j* ***in*** *S if $t_{i2} \leq t_{j1}$ and for every $T_k \in S$, either $t_{k2} \leq t_{i2}$ or $t_{k1} \geq t_{j1}$.*

Intuitively, $\mathcal{I}$ is a *coherent* interpretation of a go theory G if for each object o, there is a time interval T such that for every time point $t \in T$, $\mathcal{I}(o, t)$ either satisfies a go-atom in G or keeps the object at the destination of the last satisfied go-atom in G .

Definition 45 (Coherent Model and Theory) *Let $\mathcal{I}$ be a model of the go theory G. Let $G[o] = \{g_1, g_2, \ldots, g_n\}$ be the set of all go-atoms in G about object o. $\mathcal{I}$ is* ***coherent w.r.t.*** *o* ***and*** *G iff*

(i) There are time intervals $T_1 = [t_{11}, t_{12}]$, $T_2 = [t_{21}, t_{22}]$, …, $T_n = [t_{n1}, t_{n2}]$ such that for each i, $\mathcal{I}$ satisfies g_i over T_i and

(ii) For every pair of time intervals T_i, T_j such that T_i immediately precedes T_j in $\{T_1, T_2, \ldots, T_n\}$ the following holds:

$$\forall t \in [t_{i2}, t_{j1}] \;\; \mathcal{I}(o, t) = loc_2(g_i), \text{ i.e. destination of } g_i.$$

I is a ***coherent model of*** *G iff $\mathcal{I}$ is coherent w.r.t. o and G for all objects o.*

G is a ***coherent go-theory*** *iff G has a coherent model.*

Example 27 *Let G be the go theory in Figure 8.1. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the two interpretations in Example 26. $\mathcal{I}_1$ is coherent with respect to G and $p22$ because it satisfies g_1 over $[11, 19]$, g_2 over $[32, 40]$ and in between $[19, 32]$ plane $p22$ is in Boston. $\mathcal{I}_2$ is not coherent with respect to G and $p22$ because although it satisfies g_1 over $[10, 18]$, g_2 over $[30, 41]$ during $[18, 30]$, $p22$ does not stay in Boston which is the destination of g_1.*

We now define the concept of coherent entailment.

Definition 46 (MCWA entailment) *Let L be a ground literal and G be a go theory. G* ***entails*** *L* ***via MCWA****, denoted $G \models^{mcwa} L$, iff every coherent model of G also satisfies L.*

The MCWA is inspired by Minker's generalized closed world assumption [55] where a class of models is used to check if a given literal is true. We do the same here. The following example shows that the MCWA can handle examples such as the Planes example.

Example 28 *Let G be the go theory in Figure 8.1. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the interpretations in Example 26. Suppose $a = \mathsf{in}(p22, (75, 200), (85, 210), 23, 30)$. $G \models^{mcwa} a$ since in all coherent models of G, during $[23, 30]$ plane $p22$ is in Boston which is inside the rectangle of the atom a.*

Suppose $b = \mathsf{in}(p22, (55, 185), (80, 200), 23, 30)$. Then $G \models^{mcwa} \neg b$ since in all coherent models of G, during $[23, 30]$ plane $p22$ stays in Boston which is not in the rectangle of the atom b.

Also note that $G \not\models a$ and $G \not\models \neg b$ because according to the semantics in [89] plane $p22$ can be anywhere during $[23, 30]$.

8.3 Checking Coherence

The following lemma and definition are useful in checking whether a go-theory has a coherent model or not.

Definition 47 *Suppose G is a go theory, o is an object and π is a plan for G^o. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements of π. π is **spatially continuous** iff for every i, $1 \leq i < n$, $loc_2(\gamma_i) = loc_1(\gamma_{i+1})$, i.e., γ_i's destination is γ_{i+1}'s origin;*

Lemma 28 *A go theory G is coherent iff for every object o there is a spatially continuous plan π_o for G^o.*

The following theorem shows that checking the coherence of a go-theory is NP-complete.

Theorem 16 *Checking the coherence of a go theory is NP-complete.*

Proof: NP-hardness can be demonstrated by reducing the NP-complete problem *sequencing with release times and deadlines* [34] to coherence-checking for a go theory. Let $S = \{(t_1, r_1, d_1, \ell_1) \ldots (t_n, r_n, d_n, \ell_n)\}$ be an instance of the sequencing problem where every t_i is a task with release time r_i, deadline d_i and length ℓ_i then a go theory G induced by S is defined as follows: Arbitrarily select two points P_0, P_0' and an object o. Initially $G = \{\mathsf{go}(o, P_0, P_0', D, D, D+1, D+1, v, v)\}$ where D is the maximum of deadlines for any task in S and v is the distance between P_0 and P_0' . For any task $t_i \in S$, arbitrarily pick a point P_i such that $\mathsf{dist}(P_0, P_i) = \ell_i$ and there is no $j < i$ such that P_0,P_i and P_j are collinear. Add two atoms g_{i1}, g_{i2} to G where $g_{i1} = \mathsf{go}(o, P_0, P_i, r_i, d_i, r_i, d_i, 2, 2)$ and $g_{i2} = \mathsf{go}(o, P_i, P_0, r_i, d_i, r_i, d_i, 2, 2)$.

Membership in NP can be demonstrated by modifying the $Consistent(G)$ algorithm. As shown in Lemma 28 we need to find a spatially continuous plan for each object. Thus in $Consistent(G)$ before returning yes we need to check if the plan is spatially continuous and it should return yes, only when the plan is spatially continuous. Note that this check can be implemented in O(n) time. Hence the modified algorithm also runs in nondeterministic polynomial time.

8.4 Coherent Query Answering

Since incoherent theories entail everything the following algorithms are designed for coherent go theories. This section provides algorithms to check for MCWA-entailment of both positive and negative ground-in literals.

8.4.1 Ground in Queries

In this section, we show how to check whether a ground atom $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$ is MCWA-entailed by a *go*-theory G.

First, consider a go theory G=$\{g_1, g_2\}$ about an object o, and a ground atom $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$. Assume Figure 8.2 depicts $Vol(a)$ and the two line segments $[P_1, P_3]$, $[P_3, P_5]$ representing movements defined by g_1 and g_2. In any coherent model of G, g_1 will be satisfied before g_2. Hence the object enters $Vol(a)$ at P_2 and leaves $Vol(a)$ at P_4. If o always arrives at P_2 before t_2 and always leaves P_4 after t_1 subject to the constraints in G, then we can say that $G \models^{mcwa} a$.

For an arbitrary go theory, an object might enter and leave $Vol(a)$ multiple times.

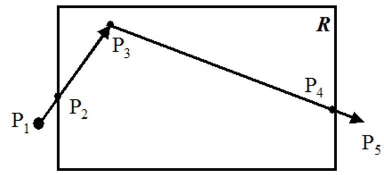


Figure 8.2: Spatial layout of two go atoms going from P_1 to P_3 and P_3 to P_5 and the volume R. In a coherent model, o stays in R between the points P_2 and P_4.

We need to identify these entrance and exit points as well as the atoms that contain them.

Definition 48 *Let L be a sequence of line segments $\ell_1 = [P_{11}P_{12}]$, $\ell_2 = [P_{21}P_{22}]$,. . ., $\ell_n = [P_{n1}P_{n2}]$ such that for $1 \leq i < n$, $P_{i2} = P_{(i+1)1}$. Let V be a cuboid volume. An* entry-exit of L for V is (i, j) *iff*

- $\ell_i \cap V \neq \emptyset$ *and* $i > 1 \implies P_{i1} \notin V$
- $\ell_j \cap V \neq \emptyset$ *and* $j < n \implies P_{j2} \notin V$
- $\forall k \in [i, j)\ P_{k2} \in V$

The following lemma gives necessary and sufficient conditions for $G \models^{mcwa} a$ when the atoms in G are satisfied w.r.t. a specific plan and the object enters and exits $Vol(a)$ multiple times.

Lemma 29 *Let G be a coherent go theory, o be an object and π be a spatially continuous plan for G^o. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements of π. Let $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$ be an atom. All instances of π that are coherent satisfy a iff there is an entry-exit (i, j) of $LS(\gamma_1) \ldots LS(\gamma_n)$ for $Vol(a)$ such that*

$$T^+(G^o, \pi, \gamma_i, P_i) \leq t_2 \text{ and } t_1 \leq T^-(G^o, \pi, \gamma_j, Q_j).$$

Algorithm $\mathsf{CheckCoherentIn}(G, \pi, a)$

1. Suppose $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$;
2. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements of π
3. **if** pi is not spatially continuous **then return** true
4. for each entry-exit (i, j) of $LS(\gamma_1) \ldots LS(\gamma_n)$ for $Vol(a)$
   - Let $[P_i, Q_i] = LS(g_i) \cap Vol(a)$ and $[P_j, Q_j] = LS(g_j) \cap Vol(a)$
   - **if** $T^+(G^o, \pi, \gamma_i, P_i) \leq t_2$ and $t_1 \leq T^-(G^o, \pi, \gamma_j, Q_j)$ **then return** true
5. **return** false

Algorithm 8.16: CheckCoherentIn algorithm

where $[P_k, Q_k] = LS(\gamma_k) \cap Vol(a)$*,*

The CheckCoherentIn algorithm (see Algorithm 8.16) uses this lemma to check for MCWA-entailment w.r.t. a specific plan.

Theorem 17 *Suppose* G *is a coherent* go*-theory and* $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$ *is a ground atom. Then:* a *is entailed by* G *via* ***MCWA*** *iff for every plan of* G^o*, the algorithm* $\mathsf{CheckCoherentIn}(G, \pi, a)$ *returns "true".*

8.4.2 Ground $\neg$in Queries

We now address the problem of checking whether $\neg\mathsf{in}(o, Q_1, Q_2, t_1, t_2)$ is MCWA-entailed by a *go*-theory G.

Consider a coherent go theory G=$\{g_1, g_2\}$ about an object o, and an atom $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$. As before Figure 8.2 depicts $Vol(a)$ and two line segments $[P_1, P_3]$, $[P_3, P_5]$ representing the movements defined by g_1 and g_2. Note that in any coherent model of G, g_1 is satisfied before g_2. Hence the object enters $Vol(a)$ at point P_2 and leaves $Vol(a)$ at point P_4. $G \models^{mcwa} \neg a$ iff

Algorithm $\mathsf{CheckCoherentNotIn}(G, \pi, \neg a)$

1. Suppose $a = \mathsf{in}(o, p_1, p_2, t_1, t_2)$
2. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements of π
3. **if** π is not spatially continuous **then return** true
4. **if** $t_1 < T^+(G^o, \pi, \gamma_1, loc_1(\gamma_1))$ **return** false
5. **if** $t_2 > T^-(G^o, \pi, \gamma_n, loc_2(\gamma_n))$ **return** false
6. **for each** entry-exit (i, j) of $LS(\gamma_1) \ldots LS(\gamma_n)$ for $Vol(a)$
   - Let $[P_i, Q_i] = LS(g_i) \cap Vol(a)$ and $[P_j, Q_j] = LS(g_j) \cap Vol(a)$
   - **if** $T^-(G^o, \pi, \gamma_i, P_i) \leq t_2$ and $t_1 \leq T^+(G^o, \pi, \gamma_j, Q_j)$ **then return** false
7. **return** true

Algorithm 8.17: CheckCoherentNotIn algorithm

- t_1 is greater than or equal to the start time of g_1 in any coherent model of G.
- t_2 is smaller than or equal to the end time of g_2 in any coherent model of G.
- Let T_1 be the earliest arrival time to P_2 and T_2 be the latest arrival time to P_4 in any coherent model of G then $T_1 > t_2$ or $T_2 < t_1$.

The following lemma gives necessary and sufficient conditions for $G \models^{mcwa} a$ when the atoms in G are satisfied w.r.t. a specific plan and the object enters and exits $Vol(a)$ multiple times.

Lemma 30 *Let G be a coherent go theory, o be an object and π be a spatially continuous plan for G^o. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements of π. Let $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$ be an atom. All instances of π that are coherent satisfy $\neg a$ iff all of the following hold:*

- $T^+(G^o, \pi, \gamma_1, loc_1(\gamma_1)) \leq t_1$
- $T^-(G^o, \pi, \gamma_n, loc_2(\gamma_n)) \geq t_2$
- $\forall$ *entry-exit* (i, j) *of* $LS(\gamma_1) \ldots LS(\gamma_n)$ *for* $Vol(a)$,

  $T^-(G^o, \pi, \gamma_i, P_i) > t_2$ *or* $T^+(G^o, \pi, \gamma_j, Q_j) < t_1$

where $[P_k, Q_k] = LS(\gamma_k) \cap Vol(a)$.

Theorem 18 *Suppose* G *is a coherent* go-*theory and* $L = \neg \mathsf{in}(o, q_1, q_2, t_1, t_2)$ *is a ground literal. Then* L *is entailed by* G *via* ***MCWA*** *iff for every plan of* G^o*, the algorithm* CheckCoherentNotIn(G, π, L) *(see Algorithm 8.17) returns "true".*

Chapter 9

Deconfliction

9.1 Motivation

There are numerous applications where we develop plans for moving objects and expect those moving objects to achieve various goals and satisfy various constraints. For instance, we have built an application with the US Navy in which motion plans for ships need to ensure that certain regions are avoided (constraints) and certain other regions are visited (goals). Likewise, air traffic controllers need to ensure that airplanes do not stray into the airspace of another airplane, while still being able to achieve their goals in a timely fashion. Go theories provide a logical framework to reason about sets of moving object plans however they do not support goals and/or constraints of the types mentioned above.

The primary goal in this chapter is to develop a theory which allows a user to examine a set of movement plans (i.e. go theories), together with a specification of the goals (e.g. to reach Seattle by 3pm) of the vehicles and any constraints (e.g. to avoid no-fly zones) and to check if the set is consistent and whether the goals (resp. constraints) are achieved (resp. satisfied). There is no conflict if the answer is "yes." On the other hand, if the answer is "no," then we must deconflict the plans — this can be done in many ways. The *deconfliction problem* is the problem of finding a way by which the plans can be reorganized so that (i) all the vehicles can achieve their goals and (ii) so that all the constraints are satisfied. A deconfliction may not always exist. This chapter presents a formal definition of the deconfliction problem, and the DECON algorithm which checks if a deconfliction exists (and if so finds one).

9.2 Integrity Constraints and Deconfliction

We will first define *ordered-go-theories* which is basically a set of go atoms and a set of total orders imposed on all atoms relating to the same object. This is an important class that covers a wide range of applications, e.g. transportation applications.

Definition 49 *Let G be a go theory and $\sqsubseteq$ be a partial order on G such that $\sqsubseteq = \cup_{o \in O} \sqsubseteq^o$ where O is the set of all objects referenced in G and $\sqsubseteq^o$ is a total order on G^o which is spatially continuous. Then $\langle G, \sqsubseteq \rangle$ is an **ordered-go-theory**.*

Throughout the rest of the chapter the notation $\sqsubseteq^o$ denotes the total order on atoms of G^o in an ordered-go-theory $\langle G, \sqsubseteq \rangle$. Also if $\Pi = \langle G, \sqsubseteq \rangle$ is an ordered-go-theory and o is an object then $\Pi^o = \langle G^o, \sqsubseteq^o \rangle$ denotes the restriction of Π to o.

Next we will define satisfaction of an ordered-go-theory:

Definition 50 *Let $\Pi = \langle G, \sqsubseteq \rangle$ be an ordered-go-theory and o be an object. Suppose $g_1 \sqsubseteq^o g_2 \cdots \sqsubseteq^o g_n$ are the atoms in G^o. An interpretation $\mathcal{I}$ satisfies Π w.r.t. o iff there are time intervals $[t_{11}, t_{12}] \ldots [t_{n1}, t_{n2}]$ such that*

- *$\forall\, g_i\ \mathcal{I}$ satisfies g_i over $[t_{i1}, t_{i2}]$*
- *$\forall\, g_i, g_j\ t_{i2} \leq t_{j1}$*

An interpretation $\mathcal{I}$ satisfies Π iff for every object o, $\mathcal{I}$ satisfies Π w.r.t. o.

Each object plan must satisfy some constraints. For example, a plane might have a goal ("Be in Boston by 5pm."). Likewise, there may be some "nogo" regions that must be avoided - moreover, these nogo regions may change with time. For example, military airspace has this feature. Likewise, shuttle launches from Cape Canaveral can temporally modify no-fly zones.

Definition 51 *An **integrity constraint** is either a goal or an nogo region where:*

- *A **goal** is an atom of the form $c = \mathsf{in}(o, P_1, P_2, t_1, t_2)$ (saying that o should be in the volume $Vol(c)$ at some time in the $[t_1, t_2]$ interval).*
- *A **nogo region** is a literal of the form $c = \neg\mathsf{in}(o, P_1, P_2, t_1, t_2)$ (saying that object o is not in the volume $Vol(c)$ at any time in the $[t_1, t_2]$ interval).*

For a plane to satisfy these constraints, there must be no possible way that the go-theory associated with it can cause the plane to either miss its goal or accidentally stray into a no go region. Therefore, these constraints must be logical consequences of the ordered-go-theory in order for them to be satisfied.

Definition 52 *A coherent ordered-go-theory $\langle G, \sqsubseteq\ \rangle$ is* safe w.r.t a set C of integrity constraints *iff for every $c \in C$, $\langle G, \sqsubseteq\ \rangle \models^{mcwa} c$.*

Note that MCWA entailment is used above so that we avoid some of the undesirable interpretations mentioned in the preceding chapter[1]. The example below illustrates the concept of safety.

[1]Coherence and motion closed world entailment definitions can trivially be modified for ordered-go-theories

Example 29 *Let* $G_1 = \{go(o, (0,0,0), (12,16,0), 3, 6, 7, 10, 4, 10)\}$ *and* $G_2 = \{go(o', (10,50,0), (90,50,0), 5, 9, 45, 49, 2, 2)\}$ *be two go theories.* $C_1 = \{in(o, (12,16,0), (12, 16,0), 7, 9)\}$ *and* $C_2 = \{\neg in(o', (30,40,0), (40,60,0), 12, 17)\}$ *are two sets of integrity constraints.*

- $\langle G_1, \emptyset\rangle$ *is not safe w.r.t.* C_1 *because there is a model* $\mathcal{I}$ *of* G_1 *which satisfies the go-atom in* G_1 *over* $[5, 10]$*, thus arriving at the goal* $(12, 16, 0)$ *too late.*
- $\langle G_2, \emptyset\rangle$ *is not safe w.r.t.* C_2 *because there is a model* $\mathcal{I}$ *of* G_2 *such that* $\mathcal{I}$ *satisfies the atom in* G_2 *over* $[5, 45]$*, which requires* o' *to be in the nogo region from time 15 to 20. Thus* $\mathcal{I}$ *does not satisfy the nogo region in* C_2*.*

Suppose $G'_1 = \{go(o, (0,0,0), (12,16,0), 3, 6, 7, 9, 4, 10)\}$ *and* $G'_2 = \{go(o', (10,50,0), (90,50,0), 8, 9, 48, 49, 2, 2)\}$ *are slightly different from* G_1 *and* G_2 *respectively.*

- $\langle G'_1, \emptyset\rangle$ *is safe w.r.t.* C_1 *because in every model* $\mathcal{I}$*,* $\mathcal{I}$ *satisfies the atom in* G'_1 *over* $[t_1, t_2]$ *such that* $7 \leq t_2 \leq 9$*.*
- $\langle G'_2, \emptyset\rangle$ *is also safe w.r.t.* C_2 *because in every model* $\mathcal{I}$*,* $\mathcal{I}(o', t)$ *is on the line segment* $[(10,50,0), (29,50,0)]$ *for every* $t \in [12, 17]$ *which is outside the nogo region in* C_2*.*

The above Example shows that when an ordered-go-theory G is not safe w.r.t. a set of integrity constraints, we may be able to "restrict" atoms in G so that safety is achieved.

Definition 53 *A* ***restriction*** *of a consistent go-atom* g *is a consistent go-atom* g' *such that:*

- $obj(g) = obj(g')$*,* $loc_1(g) = loc_1(g')$ *and* $loc_2(g) = loc_2(g')$
- $t_1^-(g) \leq t_1^-(g') \leq t_1^+(g') \leq t_1^+(g)$
- $t_2^-(g) \leq t_2^-(g') \leq t_2^+(g') \leq t_2^+(g)$

- $v^-(g) \le v^-(g') \le v^+(g') \le v + (g)$.

Note that every model of g' is also a model of g.

Definition 54 *Let $\langle G, \sqsubseteq \ \rangle$ be coherent ordered-go-theory which is not safe w.r.t. a finite set C of integrity constraints. A* ***deconfliction*** *of $\langle G, \sqsubseteq \ \rangle$ w.r.t. C is a coherent ordered-go-theory $\langle G', \sqsubseteq' \ \rangle$ such that*

- *There is a bijection f from G to G' such that if $f(g) = g'$ then g' is a restriction of g.*
- *$\langle G', \sqsubseteq' \ \rangle$ is safe w.r.t. C and*
- *If $g_1 \sqsubseteq^o g_2$, then $f(g_1) \sqsubseteq'^o f(g_2)$.*

Intuitively, deconfliction is obtained from G by replacing each atom in G by a restriction of that atom in a way that preserves order, coherence and entails the desired goals without straying into nogo regions.

Example 30 *Let G_1 , G_2, G'_1, G'_2 and C_1, C_2 be as in Example 29. Then $\langle G'_1, \emptyset \rangle$ is a deconfliction of $\langle G_1, \emptyset \rangle$ w.r.t. C_1. Then $\langle G'_2, \emptyset \rangle$ is a deconfliction of $\langle G_2, \emptyset \rangle$ w.r.t. C_2.*

9.3 A Deconfliction Algorithm

This section presents the DECON deconfliction algorithm. The algorithm is quite complex - we proceed to define the algorithm via the following steps. First, we show how to associate a set of linear constraints with each ordered-go-theory and each object and use these linear constraints for consistency checking (polynomial time). Second, we identify *spatial candidates* for a given object - these are parts of the theory that, if restricted, can potentially deconflict the theory. Third, a spatial candidate is a candidate for restriction if it also satisfies temporal requirements. Finally, the algorithm cycles through the candidates in an attempt to find one that actually works.

9.3.1 Linear Constraints

Given a go-atom g, let $\ell(g)$ be the length of $LS(g)$. Given an $\mathsf{in}()$-literal c, $Vol(c)$ is the set of all points that lay in the volume defined in c.

Suppose $\langle G, \sqsubseteq \rangle$ is an ordered-go-theory and o is an object. We associate a set $\mathcal{L}(G, o, \sqsubseteq)$ of constraints as follows. For each go-atom $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+) \in G^o$, let S_g denote the actual start time when this motion is executed, and let v_g be the reciprocal of the average velocity with which the object travels during this motion [2]. Then

- for every $g \in G^o$ $\mathcal{L}(G, o, \sqsubseteq)$ contains:
  - $t_1^- \leq S_g \leq t_1^+$
  - $1/v^+(g) \leq v_g \leq 1/v^-(g)$
  - $t_2^- \leq S_g + \ell(g) * v_g \leq t_2^+$,
- for every $g, g' \in G^o$ s.t. $g \sqsubseteq^o g'$, $\mathcal{L}(G, o, \sqsubseteq)$ contains: $S_g + \ell(g) * v_g \leq S_{g'}$.

The first two constraints are obvious. The third constraint just says that it should be possible to reach the destination during the specified interval at the average velocity of travel. The last constraint merely ensures that the $\sqsubseteq^o$ ordering is preserved. It is easy to see that $\langle G, \sqsubseteq \rangle$ is consistent iff $\mathcal{L}(G, o, \sqsubseteq)$ is solvable for all objects o.

[2] Using the reciprocal is necessary to maintain linearity of the equations

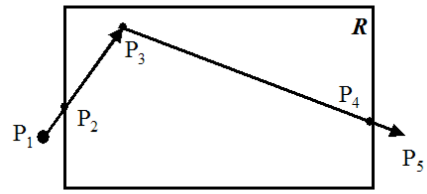


Figure 9.1: Spatial layout of two go atoms (related to object o) going from P_1 to P_3 and P_3 to P_5 and a rectangle R.

9.3.2 Spatial Candidates

Suppose G^o is unsafe. A spatial candidate identifies portions of the path of o that can be restricted further to ensure deconfliction. Let us start with an example.

Example 31 *Figure 9.1 shows two go-atoms. g_1 is a movement from P_1 to P_3, while g_2 is a movement from P_3 to P_5. Suppose the box in the figure represents a goal (i.e. we want the object to be in this box at some time). In this case, the portions of g_1, g_2 that are in the rectangle form a spatial candidate (because it satisfies the goal of being in the region). On the other hand, suppose the box represents a nogo region. In this case, the portions of g_1, g_2 that are outside the box represents a spatial candidate (because it satisfies the requirement of not being in the region).*

Formally, a spatial candidate is a four tuple that contains the segments identified above, as well as the relevant go-atoms. If C is a set of integrity constraints, then $C[o]$ denotes the constraints in C that refer to object o.

Definition 55 *Suppose $\langle G, \sqsubseteq \rangle$ is a coherent ordered-go-theory, C is a finite set of integrity constraints and o is an object. Suppose $g_1 \sqsubseteq^o g_2 \sqsubseteq^o \cdots \sqsubseteq^o g_n$ are the atoms in G^o and suppose c is a constraint in $C[o]$.*

The quadruple $\langle P, g_i, Q, g_j \rangle$ *is a* ***spatial candidate*** *for* $c = in(o, R_1, R_2, t_1, t_2)$ *w.r.t.* $\langle G, \sqsubseteq \rangle$ *iff* $g_i \sqsubseteq^o g_j$ *and:*

- $LS(g_i) \cap Vol(c) \neq \emptyset$ *and* $i > 1 \implies loc_1(g_i) \notin Vol(c)$
- $LS(g_j) \cap Vol(c) \neq \emptyset$ *and* $j < n \implies loc_2(g_j) \notin Vol(c)$
- $\forall k \in [i, j)\ loc_2(g_k) \in Vol(c)$
- P *is the closest point to* $loc_1(g_i)$ *in* $LS(g_i) \cap Vol(c)$*.* Q *is the closest point to* $loc_2(g_j)$ *in* $LS(g_j) \cap Vol(c)$*.*

Suppose c *is of the form* $\neg in(o, R_1, R_2, t_1, t_2)$ *and* $\theta_1 = \langle P_1, g_{11}, Q_1, g_{12} \rangle \ldots \theta_z = \langle P_z, g_{z1}, Q_z, g_{z2} \rangle$ *are the spatial candidates of* $in(o, R_1, R_2, t_1, t_2)$ *such that* $g_{11} \sqsubseteq^o g_{21} \ldots \sqsubseteq^o g_{z1}$*.* $\langle P, g_i, Q, g_j \rangle$ *is a* ***spatial candidate*** *for* c *iff one of the following holds:*

- $i = 1$ *and* $P = loc_1(g_i)$ *and* $P \notin Vol(c)$ *and* $\theta_1 = \langle Q, g_j, Q_1, g_{12} \rangle$ *or*
- $j = n$ *and* $Q = loc_2(g_j)$ $Q \notin Vol(c)$ *and* $\theta_z = \langle P_z, g_{z1}, P, g_i \rangle$ *or*
- $\exists \theta_k, \theta_{k+1}$ *such that* $\theta_k = \langle P_k, g_{k1}, P, g_i \rangle$ *and* $\theta_{k+1} = \langle Q, g_j, Q_{k+1}, g_{(k+1)2} \rangle$*.*

An example of spatial candidates is given below.

Example 32 *Let us return to Example 31. If the box in Figure 9.1 denotes a goal, then* $\langle P_2, g_1, P_4, g_2 \rangle$ *is a spatial candidate (representing the segment from* P_2, P_4*). If the box is a nogo region, then* $\langle P_1, g_1, P_2, g_1 \rangle$ *and* $\langle P_4, g_2, P_5, g_2 \rangle$ *are spatial candidates.*

9.3.3 Candidates

Spatial candidates do not consider time requirements. We now define "candidates" that do. For an integrity constraint $c = \pm\mathsf{in}(o, P_1, P_2, t_1, t_2)$ (positive or negated), we use

$Extent(c)$ to denote the interval $[t_1, t_2]$.

Definition 56 *Let $\langle G, \sqsubseteq \rangle$, o and $\sqsubseteq$ be as defined in Definition 55. Suppose $g_1 \sqsubseteq^o g_2 \sqsubseteq^o \cdots \sqsubseteq^o g_n$ are the atoms in G^o and $c \in C[o]$. A spatial-candidate $\langle P, g_i, Q, g_j \rangle$ for c w.r.t. $\langle G, \sqsubseteq \rangle$ is a **candidate** for c w.r.t. $\langle G, \sqsubseteq \rangle$ iff*

- $Extent(c) \cap [t_1^-(g_i), t_2^+(g_j)] \neq \emptyset$ *when c is a goal*
- $Extent(c) \subseteq [t_1^-(g_i), t_2^+(g_j)]$ *when c is an nogo region.*

The definition says that for a spatial candidate $\langle P, g_i, Q, g_j \rangle$ to be a candidate w.r.t. a goal, the extent of the goal atom must intersect the interval defined by the earliest start time of g_i and the latest end time of g_j - otherwise there is no hope. Likewise, for a nogo region, the extent of the $\neg\mathsf{in}()$ atoms should be a subset of the interval defined by the earliest start time of g_i and the latest end time of g_j. As this spatial candidate keeps the object out of the region, it now ensures it does so temporally as well.

The following result now states that if a deconfliction of a ordered-go-theory w.r.t. a set of integrity constraints exists, then there is a candidate for each integrity constraint. This means that focusing on candidates is enough to find deconflictions.

Lemma 31 *Let $\langle G, \sqsubseteq \rangle$ be a coherent ordered-go-theory and **C** be a set of integrity constraints. If there is a deconfliction of $\langle G, \sqsubseteq \rangle$ w.r.t. **C** then for every integrity constraint $c \in C$ there is a candidate w.r.t. $\langle G, \sqsubseteq \rangle$.*

9.3.4 The DECON Algorithm

The following very important theorem says that when we look for a deconfliction, it is enough to restrict go-atoms so that the departure time is a point (not an interval), the arrival point is a point (not an interval) and the velocity is fixed (not in an interval). This greatly reduces our search space.

Theorem 19 *Let $\langle G, \sqsubseteq \rangle$ be a coherent ordered-go-theory and o be an object and C be a set of integrity constraints. Suppose $g_1 \sqsubseteq^o g_2 \cdots \sqsubseteq^o g_n$ are the atoms of G^o. There is a deconfliction of $\langle G, \sqsubseteq \rangle$ w.r.t. $C[o]$ iff there is a $G_o = \bigcup_i \{go(o, loc_1(g_i), loc_2(g_i), t_{i1}, t_{i1}, t_{i2}, t_{i2}, v_i, v_i)\}$ such that $\langle G_o, \sqsubseteq \rangle$ is a deconfliction of $\langle G^o, \sqsubseteq^o \rangle$ w.r.t. $C[o]$.*

For any object o, $\mathcal{L}(G, o, \sqsubseteq^o)$ contains the precise constraints describing coherence. We need additional constraints that will ensure that the object is in one of the "candidates" defined above.

Definition 57 *Let $\langle G, \sqsubseteq \rangle$, C, o, and c be as in Definition 56. Suppose $g_1 \sqsubseteq^o g_2 \sqsubseteq^o \cdots \sqsubseteq^o g_n$ are the atoms in G^o. Let $\theta = \langle P, g_i, Q, g_j \rangle$ be a candidate for c w.r.t. $\langle G, \sqsubseteq \rangle$. If $Extent(c) = [t_1, t_2]$ then $\mathcal{C}(\theta, G, \sqsubseteq, c)$ is the set of constraints:*

- $\{X_{iP} \leq t_2 \wedge t_1 \leq X_{jQ}\}$ *when c is a goal.*
- $\{X_{iP} \leq t_1 \wedge t_2 \leq X_{jQ}\}$ *when c is a nogo region.*

where X_{iP},X_{jQ} are variables. Intuitively X_{iP} represents when o will be at point P while satisfying g_i.

The Algorithm 9.18 checks if there is a deconfliction of an ordered-go-theory w.r.t integrity constraints and an object. Using Theorem 19, it searches for a constant speed, constant departure/arrival time restriction of the theory that will entail the integrity constraints.

Algorithm DECON($\Pi = \langle G, \sqsubseteq \rangle, C, o$)

1. Let $g_1 \sqsubseteq^o g_2 \cdots \sqsubseteq^o g_n$ be the atoms of G^o
2. **if** $\exists c \in C[o]$ such that $\pi[o] \models^{mcwa} \neg c$ **then return** NO
3. Let $\mathcal{D} = \mathcal{L}(G, o, \sqsubseteq)$ and $\mathcal{M} = \emptyset$
4. **for each** $c \in C[o]$ such that $\Pi[o] \not\models^{mcwa} c$
   - **If** there is no candidate for c w.r.t π **then return** NO
   - Nondeterministically select a candidate θ for c
   - Insert $\mathcal{C}(\theta, \Pi, c)$ into $\mathcal{D}$ and $\mathcal{M} = \mathcal{M} \cup \{(c, \theta)\}$.
5. **for each** variable $X_{iP} \in \mathcal{D}$
   - add $X_{iP} = S_{g_i} + v_{g_i} * \mathsf{dist}(P, loc_1(g_i))$ to $\mathcal{D}$
   - **if** $\mathcal{D}$ has a solution **return** YES
   - **else return** NO

Algorithm 9.18: DECON algorithm

We can modify the DECON algorithm to construct a deconfliction via the lemma below.

Lemma 32 *Let $\langle G, \sqsubseteq \rangle$ be a coherent ordered-go-theory and o be an object. Suppose $g_1 \sqsubseteq^o g_2 \cdots \sqsubseteq^o g_n$ are the atoms of G^o. Let ϕ be a solution of constraints $\mathcal{D}$ associated with a trace of* DECON$(\langle G, \sqsubseteq \rangle, C, o)$ *that returns "YES". Suppose $G_o = g'_1 \sqsubseteq_o g'_2 \cdots \sqsubseteq_o g'_n$ where $g_i = go(o, loc_1(g_i), loc_2(g_i), t_{i1}, t_{i1}, t_{i2}, t_{i2}, V_i, V_i)$ and*

- $V_i = 1/\phi(v_i)$
- $t_{i1} = \phi(S_i)$
- $t_{i2} = t_{i1} + \phi(v_i) * dist(g_i))$.

Then $\langle G_o, \sqsubseteq_o \rangle$ is a deconfliction for $\langle G^o, \sqsubseteq^o \rangle$ w.r.t $C[o]$.

Finally we state that the algorithm DECON is sound and complete.

Theorem 20 *$\langle G, \sqsubseteq \rangle$ has a deconfliction w.r.t. C iff for every object o such that $G^o \neq \emptyset$, DECON($\langle G, \sqsubseteq \rangle, C, o$) returns "YES" for some nondeterministic trace.*

Chapter 10

Implementation

10.1 User Interface

We have built a prototype system called LOM in Matlab a mathematical programming language. The LOM system provides a user interface to input and query go-theories.

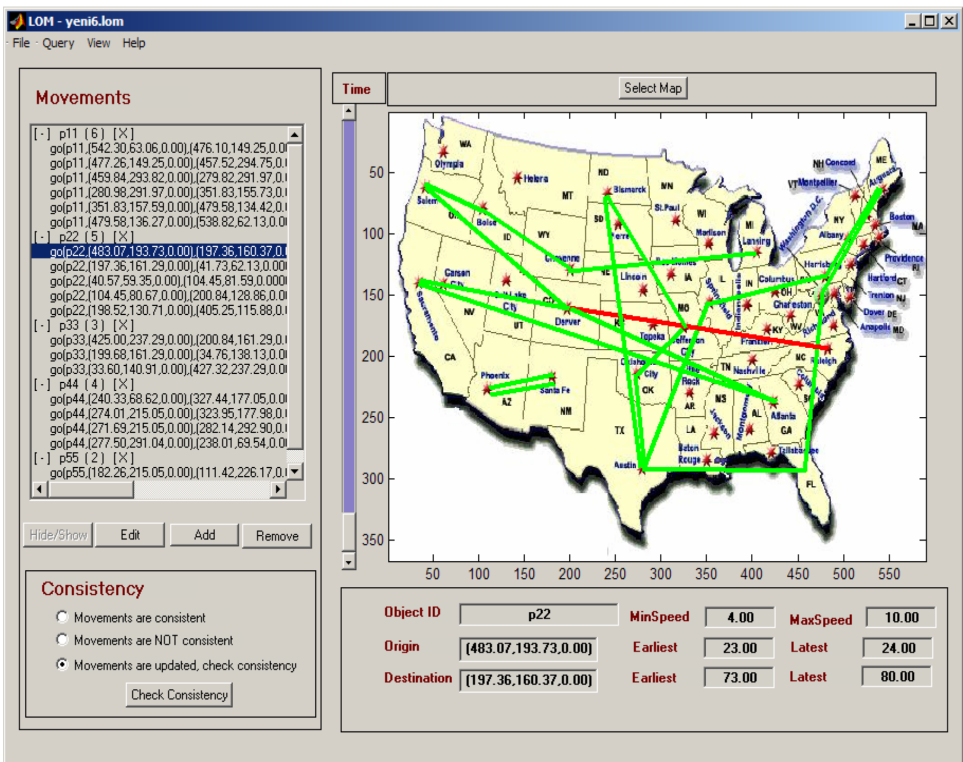


Figure 10.1: A screen shot of LOM user interface - Theory input view.

The interface has two views. The first view (which is seen in Figure 10.1) allows the user to load a map and enter the paths in of the go atoms using mouse clicks. Alternatively the user may load a previously saved theory. The atoms are organized with respect to the objects they refer to. The user may select to display these atoms on the map or not. This allows simplified views for the theory. It is possible to save the created theory. In this view the user can check the consistency of the theory that is being updated or created.

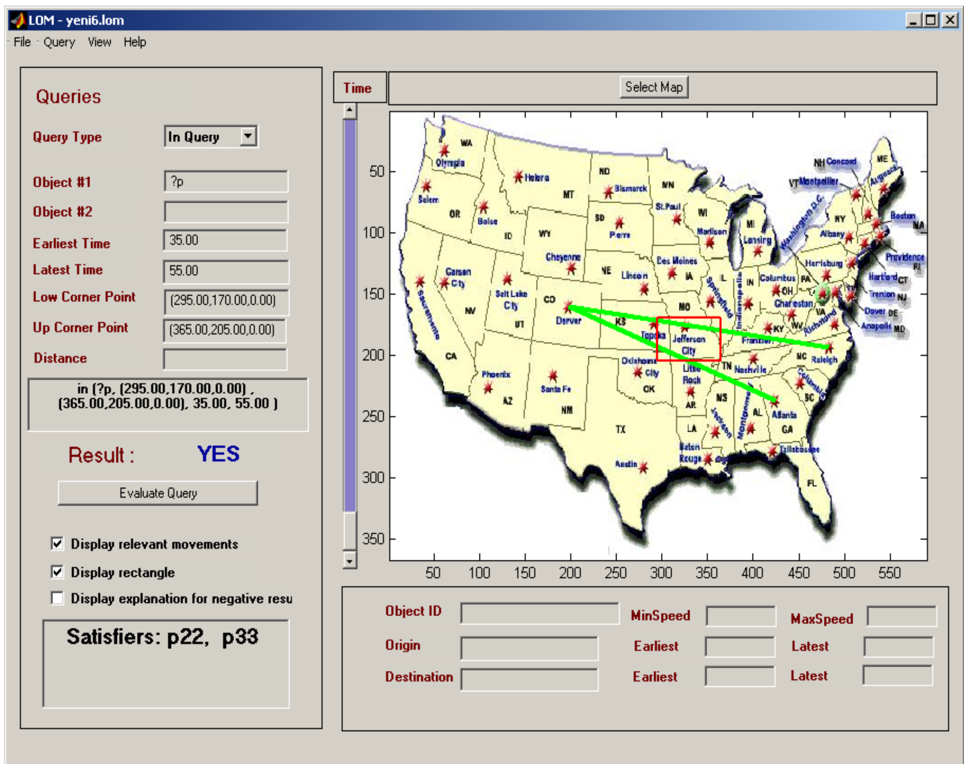


Figure 10.2: A screen shot of LOM user interface - Query view.

The second view (which is seen in Figure 10.2) allows the user to load/save or create queries and evaluate then w.r.t. the currently loaded go-theory. For non-ground theories the satisfiers are displayed at the left-bottom corner. If the related movements option is selected, only the atoms that entail the query are displayed on the map.

10.2 Consistency Checking Performance

This section presents the performance of consistency checking algorithm for simple go theories. The experiments are conducted on a Pentium 4 (3.80GHz) processor running under Windows XP and with 2GB of memory. The algorithm is implemented in Matlab.

The theories are created randomly with the following parameters: All atoms are within a rectangular region 500 by 600, the maximum speed of any object is less than 50, the difference between earliest and latest start time for any go atom is less than 10. In order to assure the randomly created theories are simple we enforced that each generated atom starts after the end time of previously generated atom.

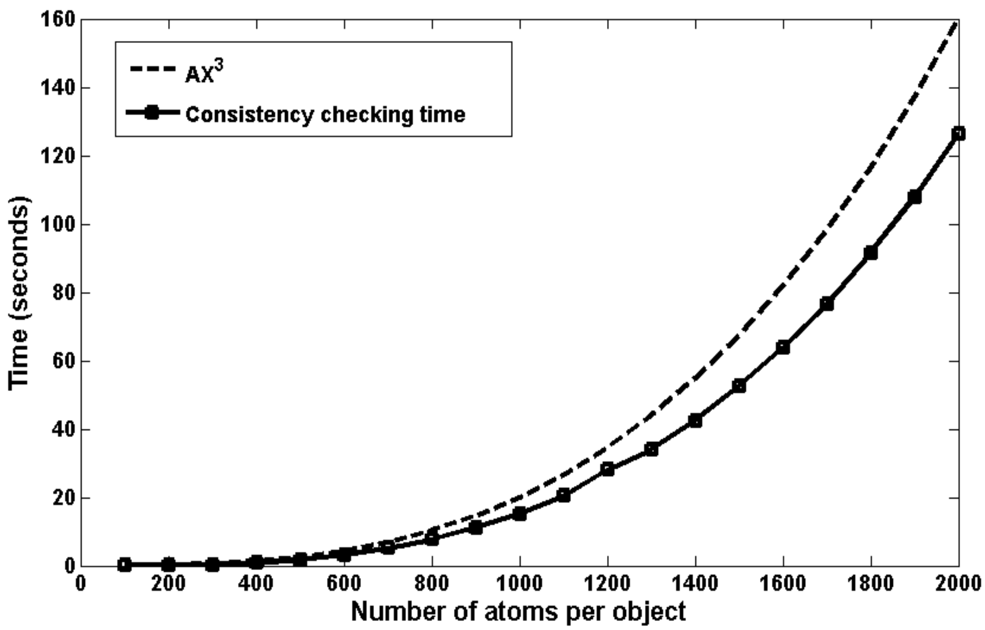


Figure 10.3: Running time for CheckSimpleConsistency

Figure 10.3 presents the running time for `CheckSimpleConsistency` algorithm for up to 2000 atoms per object in the theory (theories contain atoms about a single

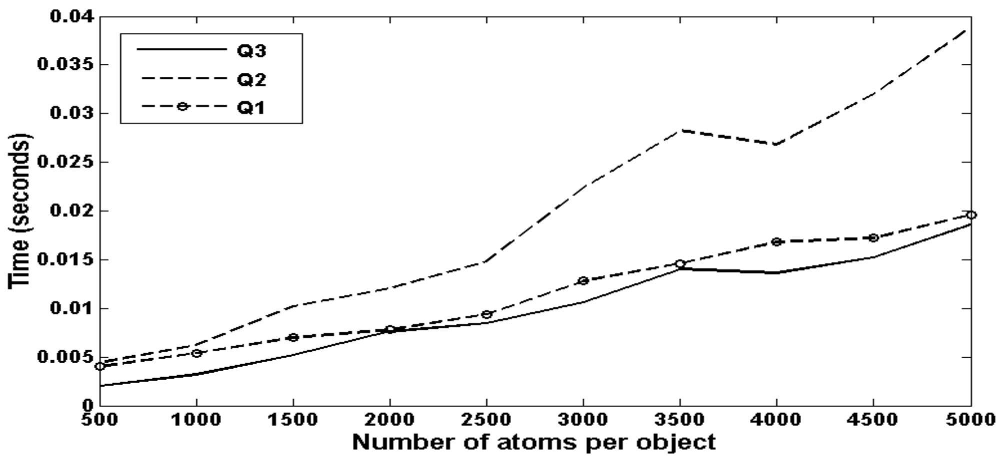


Figure 10.4: Time to answer ground in-queries $Q1, Q2$ and $Q3$

object). The data points are an average of 30 runs. As the theoretical analysis predicted the running time of the algorithm is $O(n^3)$ where n is the number of atoms in the theory. For 2000 atoms, the time to check consistency is around 130 seconds. Although this number is high in a real-world application we do not expect to have 2000 movements on a single object. For example consistency checking on a theory containing only 200 atoms, which is more likely to happen in an application, takes only milliseconds.

10.3 Ground Query Experiments- Efficiency

We have tested the performance of the ground query answering algorithms for simple theories. The experiments are conducted on a mobile Athlon XP 1800 processor with 256 MB memory running Windows XP. The simple theories are generated using the same parameters described in the previous section. We used several different query templates to test the algorithms. The following summarizes our results.

- **In queries:** $\mathsf{in}(o, R, T)$ We investigated the effect of query time interval and query rectangle size on performance of in queries. The following rectangle and intervals are used for this experiment: (1)Q1: R is 400 by 500 rectangle and T is a time interval of size $horizon/4$ where horizon is the difference between the earliest start time of any atom and latest end time of any atom. (2)Q2: R is 100 by 100 rectangle and T is a time interval of size 50 (3) Q3: R is 10 by 10 rectangle and T is a time interval of size 10. Figure 10.4 displays the running time of CheckIn for theories containing up to 5000 atoms per object. The data points are an average of 50 runs. As seen in Figure 10.4, $Q3$ takes the least time because the size of the query rectangle and query interval are too small and the algorithm quickly realizes this. $Q1$ takes slightly more than $Q3$ for a different reason. This time the rectangle and the interval are too large and the algorithm quickly returns a positive answer. The middle case, represented by $Q2$, takes longer than $Q1$ and $Q3$ because there are many related movements to the query and none of them leads to a trivially positive or negative result. Hence the algorithm has to examine almost all of the related movements.

- **Near vs. Far queries:** This experiment has two goals: 1) show the efficiency of ground and near entailment algorithms 2) show that the computation of a far-atom entailment is slightly higher than near-atom entailment when the atoms have the same parameters. We used two queries: $Q_n = \mathsf{near}(o, o', 100, 5)$ and $Q_n = \mathsf{far}(o, o', 100, 5)$ and evaluated them on the same go theories. Figure 10.5 confirms the second hypothesis. This is due to the fact that once related movements are iden-

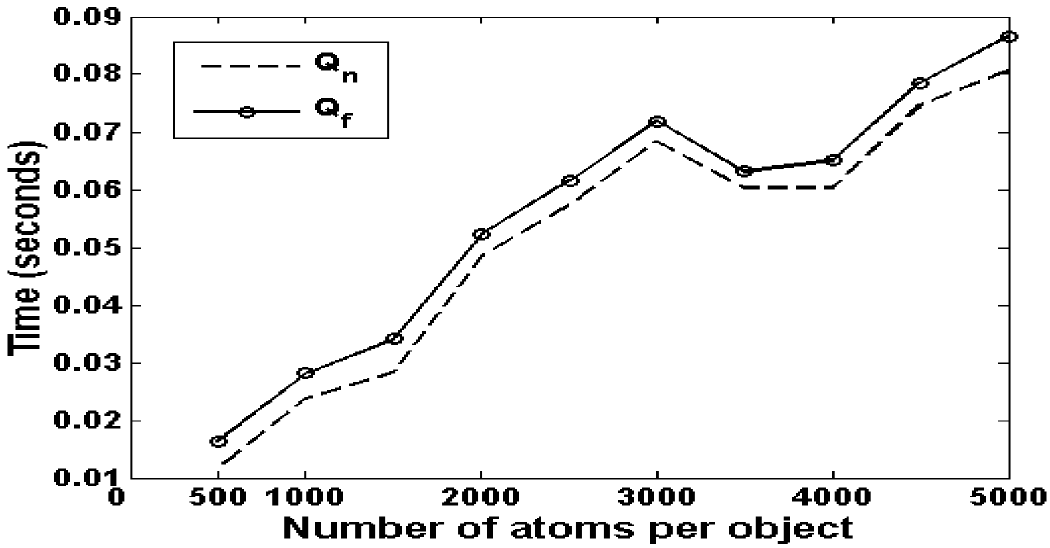


Figure 10.5: Running time for two `near` and `far` queries

tified near algorithm only checks the end points of the the space envelope whereas far algorithm has to build the neighborhood volume and perform intersection operation. Nevertheless both algorithms behave mostly similar because the dominating factor is the time spent in identifying the movements that are related to the queries. Thus the results can be further improved by using some spatio-temporal indexing. For theories containing up to 5000 atoms both `near` and `far` entailment algorithms take less than 0.9 seconds, proving the efficiency of the algorithms.

10.4 Non- Ground Query Experiments- Efficiency

We have tested the performance of our algorithms on a mobile Athlon XP 1800 processor with 256 MB memory running Windows XP.

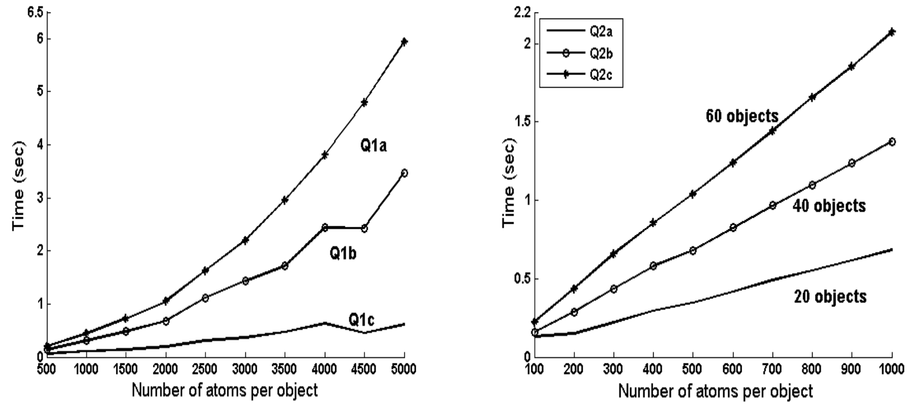
When generating simple go-theories, we ensured that for any two go atoms g_1 and g_2 about the same object, either $t_1^-(g_1) \geq t_2^+(g_2)$ or $t_1^-(g_2) \geq t_2^+(g_1)$ — this is a sufficient condition for being a simple go-theory. The following parameters control the data generation:

- Number of objects: Up to 100.
- Number of atoms per object: Up to 5000.
- Entire Region: A rectangular region of size 5000×6000.
- Local Region: A rectangular region of size 100×200 within the entire region — all go atoms for the same object are in a given local region.
- Earliest Start Time: A time interval from 0 to 5000 such that for any object and any go atom g, $0 \leq min(t_1^-(g)) \leq 5000$.
- Uncertainty: Maximum value of $t_1^+(g) - t_1^-(g)$ for any go atom.
- Speed: Maximum value of $v+$ for any go atom.

10.4.1 go Queries

We used the following query templates:

- Q1:$go(o1, ?P_1, ?P_2, t_1, t_2, t_3, t_4, ?v_1, ?v_2)$ – *Given the start and time constraint find all line segments along with the speed constraints such that object o_1 will be traveling on.* Figure 10.6a displays the running time in seconds vs. number of atoms per object for 3 variations of Q1 where the $t_4 - t_1$ changes. As the length of the time interval increases the running time of the query increases. This is due to the fact that the number of go atoms relevant to the query increases. For 5000 atoms, the query takes approximately 6 seconds to complete.



(a) **(b)**

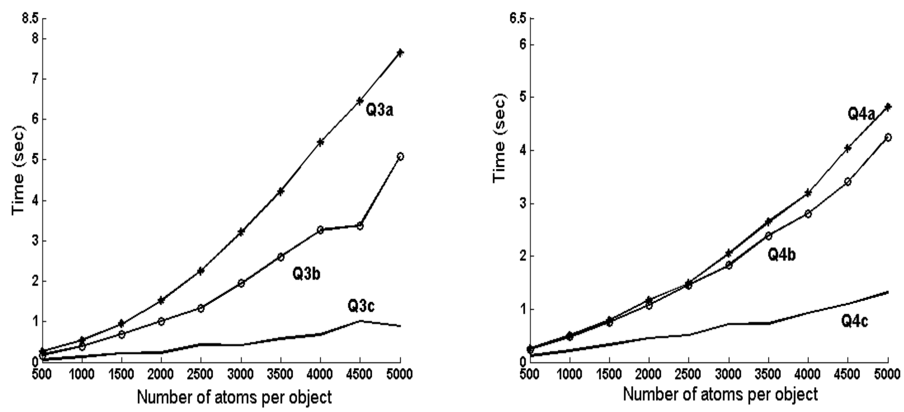
Figure 10.6: **(a)**Running time for three Q2 queries Q2a, Q2b and Q2c with time intervals $[t_{1a}, t_{2a}] \supset [t_{1b}, t_{2b}] \supset [t_{1c}, t_{2c}]$ respectively. **(b)**Running time for query Q3 for 20,40 and 60 objects.

- Q2:$go(?o, p1, p2, t1, t2, t3, t4, v1, v2)$ – *Find all objects that travel from P1 to P2 with the given time and speed constraints*. Figure 10.6b displays the running time in seconds vs. number of atoms per object for 3 cases where the number of objects in the theories vary. Clearly, the running time of Q2 is linear in the number of atoms and in the number of objects. For 1000 atoms and 60 objects, the running time is a little over 2 seconds.

10.4.2 in Queries

We used the following query templates: for in queries

- Q3:$in(o_1, ?R, t_1, t_2)$– *Find all minimal rectangles such that the object o_1 will be in that rectangle at some time in* $[t_1, t_2]$. Figure 10.7a displays the running time in seconds vs. number of atoms per object for 3 variations of Q3 where the value of $t_2 - t_1$ changes. As the length of the time interval increases, the running time of the query increases. This is due to the fact that the number of go atoms relevant to the query increases. For 5000 atoms, the query takes approximately 8 seconds to complete.



(a) **(b)**

Figure 10.7: **(a)**Running time for three Q3 queries Q3a, Q3b and Q3c with time intervals $[t_{1a}, t_{2a}] \supset [t_{1b}, t_{2b}] \supset [t_{1c}, t_{2c}]$ respectively. **(b)**Running time for three Q4 queries Q4a, Q4b and Q4c with rectangles $R_a \supset R_b \supset R_c$ respectively.

- Q4:$in(o_1, R, ?t_1, ?t_2)$– *Find all minimal time intervals such that at some time in that interval, o_1 will be in R.* Figure 10.7b displays the running time in seconds vs. number of atoms per object for 3 variations of Q4 where the size of R changes. As the size of the rectangle increases the running time of the query also increases. This is because the number of go atoms spatially relevant to the query increases.

For 5000 atoms the query takes approximately 5 seconds to complete.

- Q5:$in(?o_1, R, t_1, t_2)$– *Find all objects that will be in R at some time in $[t_1, t_2]$.* Figure 10.8 displays the running time in seconds vs. number of atoms per object for Q5 with 20,60 and 100 objects. The figure includes the running time for same query and same number of objects discussed earlier - however, we discuss index utilization as well. The figure shows that when no index is used, the running time is linear in the number of atoms and number of objects. However, when an index is used, it takes very low order constant time for the query to run. For 100 objects and 1000 atoms per object, it takes 16 seconds without an index, and under a second with the index.

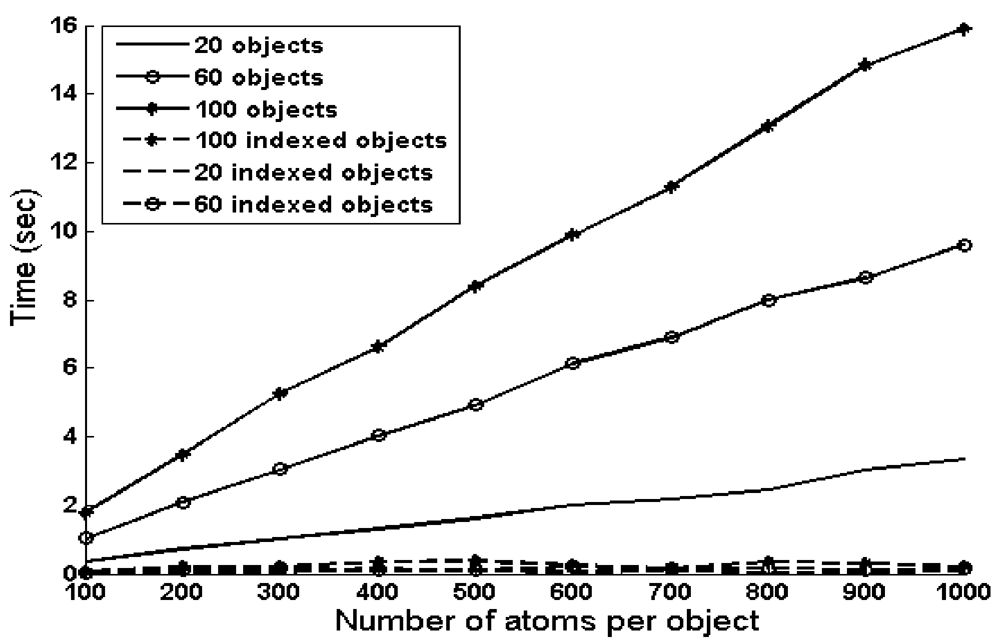
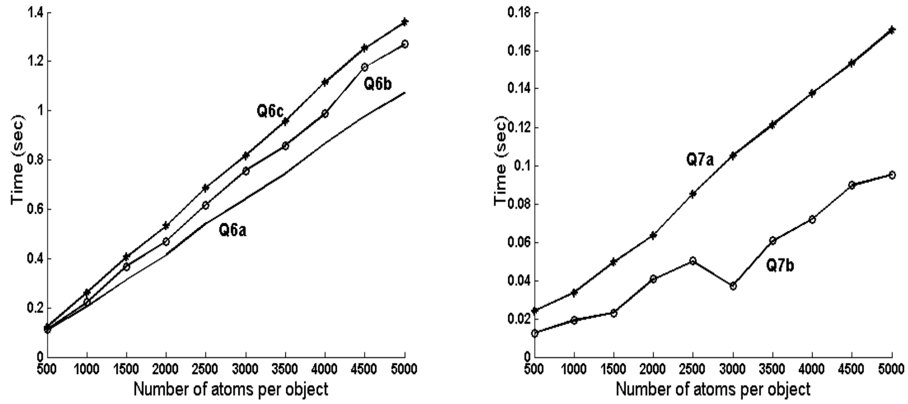


Figure 10.8: Running time for query Q5 for different number of objects and with/without index.

10.4.3 near Queries



(a) **(b)**

Figure 10.9: **(a)**Running time for three Q6 queries Q6a Q6b and Q6c with distance values d_a, d_b and d_c respectively where $d_a \geq d_b \geq d_c$. **(b)**Running time for two Q7 queries Q7a and Q7b with time intervals $[t_{1a}, t_{2a}]$ and $[t_{1b}, t_{2b}]$ respectively where $[t_{1b}, t_{2b}]$ is a certainty interval for both o_1 and o_2 and $[t_{1a}, t_{2a}]$ is not.

We used the following templates:

- Q6: $near(o_1, o_2, d, ?t_1, ?t_2)$– *Find all time intervals during which the distance between objects o_1 and o_2 is at most d.* We ran this query for three different values of d and examined how the running time of the query changed with respect to the number of atoms per object and the value of d. Figure 10.9a displays the running time in seconds vs. number of atoms per object for Q6 with with different d values. For all cases, the running time is linear with the number of atoms per object. An interesting result is that as d gets smaller, the running time increases slightly. An explanation for this is that when d is large, the the maximum distance between objects will always satisfy the nearness requirement and hence there is less computation. When d is small, the algorithm needs to solve a set of quadratic equations to determine if/when the closeness constraints will be satisfied. Despite this, the running time for the query Q6 is under 1.6 seconds when there are 5000 atoms per object.

- $Q7 : near(o_1, o_2, ?d, t_1, t_2)$– *Find the maximum distance between objects* o_1 *and* o_2 *during* $[t_1, t_2]$. We ran this query for two different values of t_1 and t_2: one that has only ∞ as a solution and one that is a certainty interval for both of the objects hence has a different solution then ∞. Figure 10.9b displays the running time in seconds vs. number of atoms per object and shows that running time is approximately linear in the number of atoms per object. When $[t_1, t_2]$ is not a certainty interval for either object, the query time is longer. In this case, the algorithm examines all atoms for the two objects and realizes that the only solution is ∞. In the second case, the algorithm terminates as soon as a pair of relevant atoms is discovered: this usually takes less time then the first case. Even for the first case, the query Q7 takes only 0.18 seconds for 5000 atoms.

- $Q8 : near(o_1, ?o_2, d, t_1, t_2)$ – *Find all objects that are at most* d *units from* o_1 *during the interval* $[t_1, t_2]$. We ran this query for 20,40 and 60 objects both with and without the indexing. Figure 10.10 displays the number of atoms per object vs running time in seconds for Q8 with with different number of objects and indexing. The reader can verify that running time is linear with respect to number of objects and number of atoms when no index is used. In this case query Q8 takes 0.7 seconds when there

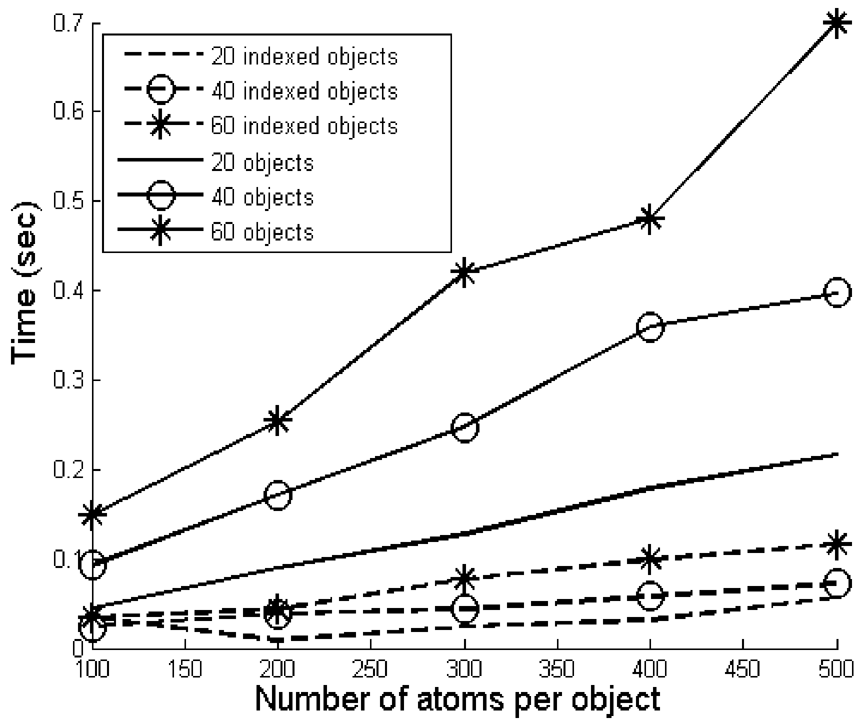


Figure 10.10: Running time for query Q8 for different number of objects and with/without indexing

are 60 objects and 500 atoms per object. On the other hand when the index is used, the same query takes under 0.1 seconds.

10.5 Motion Closed World Experiments: Efficient query answering

Determining MCWA-entailment is co-NP complete because the number of orderings spatially continuous w.r.t. G^o can be exponential. However, in the real world, we expect a go-theory to allow only a small number of orderings compatible with G^o. In other words, the respective order of movements an object is going to perform is mostly known. For example we might not know exactly when the plane $p22$ will land but we usually know where it is going to fly next. Thus, in practice there is a bound on the number of compatible total orderings per object.

For our experiments we generated random go theories with at most 256 spatially continuous orderings. This is not a hard-coded limit of our implementation. Generating random go-theories such that more than one spatially continuous ordering exists is a little bit tricky. Here is one method to generate a go theory $G = \{g_1, g_2, g_3, g_4, g_5\}$ with two spatially continuous orderings.

- Randomly pick points P_1, P_2, P_3 and P_4
- Set $loc_1(g_1) = P_1$ and $loc_2(g_1) = P_2$,
- Set $loc_1(g_2) = P_2$, $loc_2(g_2) = P_3$

  and $loc_1(g_3) = P_3$, $loc_2(g_3) = P_2$,
- Set $loc_1(g_4) = P_2$, $loc_2(g_4) = P_4$

  and $loc_1(g_5) = P_4$, $loc_2(g_5) = P_2$,
- Set temporal and speed intervals of every g_i so that g_1 is always first and the rest can be done in any order.

We have generalized the reasoning above to create random go theories with an arbitrary bound on the number of spatially continuous orderings.

We have implemented the two algorithms CheckCoherentIn and CheckCoherentNotIn in Matlab and conducted experiments on a mobile Athlon XP 1800 processor running under Windows XP and having 256MB of memory. Figure 10.11 shows the computation time of four types of queries for coherent go theories with at most 256 spatially continuous orderings and have the following properties: all points are selected randomly

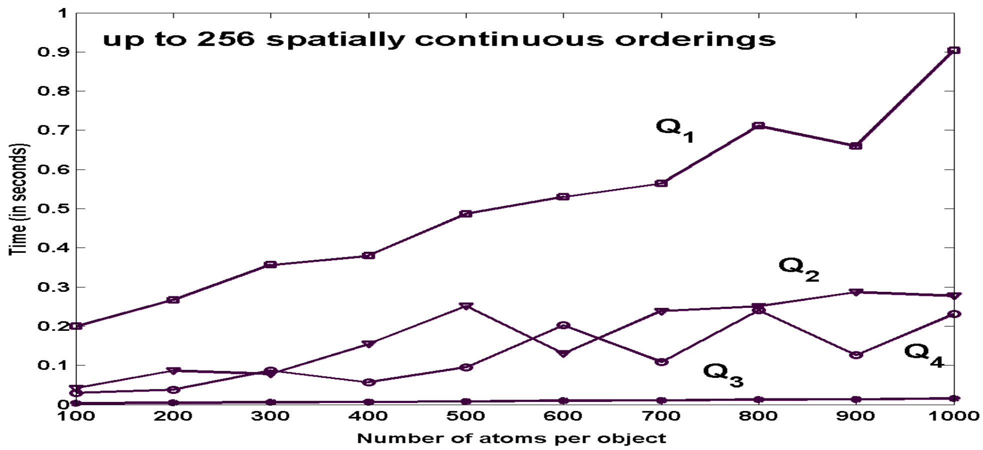


Figure 10.11: Time to answer queries $Q1, Q2, Q3$ and $Q4$ when total number of spatially continuous orderings is at most 256.

from the rectangle $[(0,0),(1000,1200)]$ and the speeds allowed for any object less than 100. The four query templates we used are:

Q1: $\mathsf{in}(o,(500,500),(550,600),0.5*h,0.75*h)$

Q2: $\mathsf{in}(o,(100,150),(350,400),h-100,h-10)$

Q3: $\neg Q1$

Q4: $\neg Q2$

where h is the latest end time for any atom related to o in the given theory. The data points in Figure 10.11 are an average of 300 runs.

The implementation performs very well, executing most queries in less than 0.3 seconds even when there are as many as 1,000 go-atoms per object. In the query $Q1$ where CheckCoherentIn returns true in almost every compatible orderings the algorithm runs in linear time with respect to number of atoms per object and takes up to 0.9 seconds

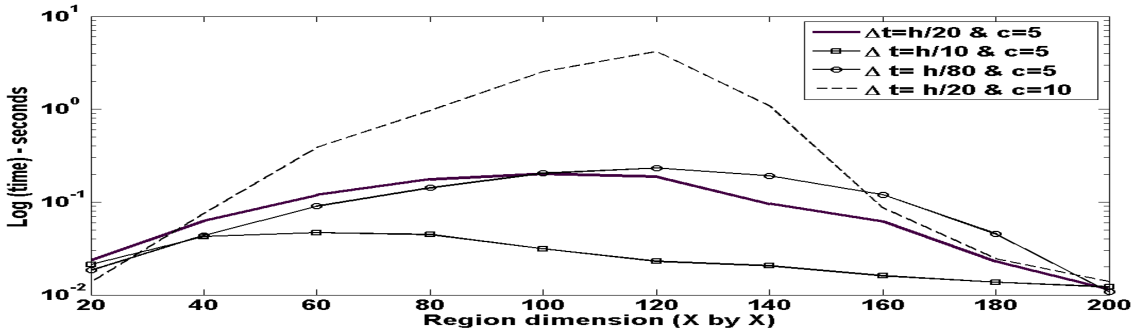


Figure 10.12: Running time of DECON for 5 and 10 integrity constraints with different time intervals and varying region sizes. h is the horizon and c is the number of constraints.

when there are 1,000 go-atoms per object. Consequently $Q3$, the complement of $Q1$, takes almost no time because CheckCoherentNotIn returns false for any compatible ordering.

10.6 Deconfliction Experiments

We implemented a 2D- version of DECON in Matlab. We conducted the experiments on a Pentium 4 (3.80GHz) processor running under Windows XP and with 2GB of memory. To investigate the behavior of DECON we created random coherent go theories and tested the algorithm with varying number of integrity constraints, with different region sizes and time intervals. Our go-theories contain 50 atoms referring to the same vehicle and each atom lies in a 200 by 200 rectangle. The maximum speed is 20 and the difference between the latest start time and earliest start time in any atom is 100.

In our experiments we varied the region size from 20 by 20 to 200 by 200. We define horizon of a theory to be the difference between latest end time for any atom and earliest start time for any atom. Figure 10.12 shows different curves for the different time intervals which ranges from $horizon/80$ to $horizon/10$ and curves for different number of constraints. The points on the graphs are an average of 500 runs. The constraints were labeled as goals or nogo regions randomly. The following summarizes our results:

- As expected increasing the number of constraints, increases the running time exponentially. DECON finds a deconfliction under 5 seconds with 10 constraints and under 0.25 seconds with 5 constraints.
- When the region size is too small or too large checking for deconfliction takes almost no time. The problems become harder when the region size is between 60 by 60 to 140 by 140 depending on the size of the constraint interval.
- When the time interval in the constraints is too large (e.g. $horizon/10$), the running time is always less than 0.05. Otherwise time to find a deconfliction depends on the size of the region and the length of time interval.

Chapter 11

Related Work

11.1 Spatio-Temporal logic

There has been extensive work on spatio-temporal logics in both the AI community and the philosophy community. Many of these are a product of combining a spatial logic, such as $RCC-8$ [24], $BRCC-8$ [85] and $S4_u$ [10], with propositional temporal logics ($\mathcal{PTL}$). The work on spatioi-temporal reasoning is mostly qualitative [4, 54, 87, 18], and focuses on relations between spatio-temporal entities while dealing with discrete time. In contrast the work in this thesis is heavily continuous and rooted in a mix of geometry and logic rather in just logic alone.

The following paragraphs summarize these spatial logics and the the temporal logic ($\mathcal{PTL}$).

- RCC-8 is a tractable subset of a more general spatial representation called RCC. In RCC regions are are primitives. Given two regions x and y, the relation $C(x, y)$ is true if the closures of the regions share a point. Intuitively $C(x, y)$ says that the regions x and y are connected. A formal semantics of RCC has been presented in [37, 78]. In RCC-8 the regions are closed and the connectivity relation leads to eight jointly exhaustive and pairwise disjoint relations. These relations are illustrated in Figure 11.1.

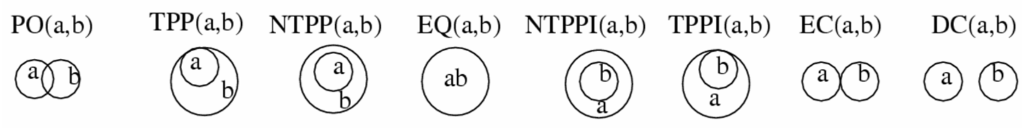


Figure 11.1: RCC-8 relations

Just like Allen's temporal interval algebra, reasoning with RCC-8 can be achieved via composition [3] or transitivity [28] tables. Thus we can deduce given $R_1(a, b)$ and $R_2(b, c)$ whether or not $R_3(a, c)$ holds. The satisfiability of RCC-8 is NP-complete [70].

- BRCC-8 is an extension of RCC-8 with boolean region terms, i.e. combination of region variables using intersection, union and complement. The computational behavior of BRCC-8 is similar to RCC-8 [86].

- $S4_u$ is an extension of the modal logic S4 [60, 35] interpreted over topological spaces. Besides the boolean connectives and region variables, there are two modal operators: I (necessary) and C(possible). S4 is a logic of topological spaces when I is the interior and C is the closure operator. $S4_u$ adds existential and universal quantifiers to S4. $S4_u$ contains both RCC-8 and BRCC-8 as fragments. The satisfiability of S4 formulas in topological models is PSPACE-complete [59, 5].

Before moving into the hierarchy of spatio-temporal logics we will briefly look at the propositional temporal logic PTL whose fragments are often used as the temporal component of these spatio-temporal logics.

PTL is a point based linearly ordered logic interpreted over various flows of time such as $\langle N, < \rangle$, $\langle Z, < \rangle$ and $\langle R, < \rangle$. The PTL formulas contain propositional variables, boolean operators ($\neg,\ \wedge,\ \vee$) and two temporal operators U (until) and S (since). Using U and S we can other important operators such as *sometimes* and *always*. It was shown in [76, 72, 71] that reasoning with PTL is PSPACE-complete over $\langle N, < \rangle$, $\langle Z, < \rangle$ and $\langle R, < \rangle$. Reasoning with a weaker fragment of PTL, $PTL_{\Box}$ that contains only *sometimes,always* and their counter parts as operators but leave out the original operators S and U is only NP-Complete [84, 76].

Here is a summary of spatio temporal logics based on the spatial logics discussed earlier and PTL as presented in [33, 32, 88].

- PST is a combination of $S4_u$ and PTL. PST is undecidable.
- ST_2 is a combination of BRCC-8 and PTL. ST_2 is EXPSPACE-complete.
- ST_2^- is a combination of RCC-8 and PTL.ST_2^- is in EXPSPACE.
- ST_1 is a combination of BRCC-8 and PTL but temporal operators can only be applied to BRCC-8 formulas and only *next time operator* (deriven from S and U) can be applied to region terms. ST_1 is EXPSPACE-complete.
- ST_1^- same as ST_1 except instead of BRCC-8 we have RCC-8. ST_1^- is PSPACE-complete.
- ST_0 s a combination of BRCC-8 and PTL but temporal operators can only be applied to BRCC-8 formulas, no operator can be applied to region terms. ST_0 is PSPACE-complete.

Despite much work on qualitative spatio-temporal theories very little work has been done on motion in such frameworks. [56, 57] describes a first order logic for reasoning about motion in a qualitative framework. It uses spatio-temporal primitives (RCC-8 and interval algebra) as the building blocks. It axiomatizes the continuity of motion and the expressive power of the theory allows for the definition of complex motion classes such as *leave, hit, reach, cross, external* and *internal*. However as [17] Muller's work allows temporal pinching, i.e. a history can disappear and reappear instantaneously. To address this [17] introduces firm connectedness and provide a hierarchy of conceptual neighborhood diagrams. These works however are purely symbolic hence has a different nature than our work.

Logics of Metric Spaces, a quantitative logic to reason about distances is introduced in [48]. Using this logic we can express statements such as: *"The distance between X and Y should be less than D."* or *"There should be a shopping plaza closer than 1 mile but not far than 3 miles to my house."*. It has shown in [48] the first order language and the two variable fragment of it that requires the triangular equality to hold is undecidable. Weaker two variable fragments of this logic is shown to be NExpTime complete. In an effort to embed temporal reasoning into metric logics, *dynamic metric logics* are introduced in [6]. In dynamic metric logics, it is possible express statements of the form "Starting form a point in P one can never reach a neighborhood of some unsafe region Q". Reasoning with dynamic metric logics is decidable but the decision problem is not elementary. The major difference between our work and dynamic metric logics is we are not using distances as our primitives and we have finitely many objects. Furthermore we explicitly model the movement of objects and unlike dynamic metric logics we deal with temporal and speed uncertainties.

There are also some work on different aspects of motion such as:

- [66] focuses on relative position and orientation of objects with existing methods for qualitative reasoning in a Newtonian framework. We are interested in identifying all objects within a given geometric region at a given point in time or within a given distance of another object at a given point in time. We also allow nondeterminacy and uncertainty via the use of time intervals and velocity intervals.

- [75] discusses the frame problem when constructing a logic-based calculus for reasoning about the movement of objects in a real-valued co-ordinate system. It reasons about the change of the position of the objects holding a certain shape through $Move(object, to-location)$ actions. There is no explicit representation of time and other than continuity the work does not specify any characteristics of the motion. Our work on the other hand explicitly represents time, speed and related uncertainty. Moreover our go atoms commits the object on a specific path (a straight line between source and destination).

- Representing vagueness and uncertainty in spatial reasoning is investigated in a series of papers [19, 20, 21, 22, 23]. These work introduce a new primitive called "crisp" for explicitly identifying the crisp boundaries of the regions and give axiomatizations for the new predicate. Furthermore they describe egg-yolk calculus to reason with vague regions. In all these work the term uncertainty is used for the boundary of the regions where as in our work, uncertainty is present at temporal and speed reasoning.

11.2 Spatio-Temporal Databases

The database literature includes work on moving-object databases, from the point of view of developing data structures to specify where objects are and what their current velocity and efficient query algorithms. The two approaches are modeling moving points and moving regions [30, 38, 31, 77, 14].

Among those the MOST data model [77] capture the continuous movement of the objects as points by defining location as a dynamic attribute that change with time even though the DB is not explicitly updated. In other words instead of the location of the object the velocity vector of the object is stored. This property enables the system to query about future states as well as the past states. However the validity of the answers concerning future are contingent on objects keeping their original velocity vector. Thus the results are volatile. This work is extended to MOD data model in [83, 82] to handle positional uncertainty of the objects. The main idea is to define a circular region around the object with a preset uncertainty radius and assume the object can anywhere in this circle. The method however does not handle speed or temporal uncertainties as we do in `go`-atoms

The work in [81] also handles the uncertainty in the trajectory of the objects. They model trajectories as 3D cylindrical bodies (2 dimensions for space and one dimension for time) which represents a 3D poly-line for path along with a threshold of deviation from the path. The representation requires exact start and end times for the path and not able to specify a speed interval for the motion.

There are many differences between these works and ours: They do not provide a formal model theory, and they do not worry about consistency because they always record observed information about where the vehicles were observed in the past (presumably such a theory is consistent because if the vehicle was at one location at time t and another location at time t' there was a physical way for it to get from the first location to the second). These work do not allow uncertainty about starting times, ending times, and velocities as our framework does.

Constraint databases [44] can be seen relevant to our work since $\mathtt{go}$-atoms temporally and spatially pose constraints on the object. Constraint databases have been applied to the temporal constraints ([9, 15]), and to the spatial constraints [62] separately. The integrated application for modeling moving objects has been studied in [79] where the paths are estimated as piecewise line segments. Also using differential geometry the relationships between objects are identified in terms of position and velocity (such as moving toward each other, in opposite directions, catching up each other etc.). This work does not address the temporal and speed uncertainties as go-atoms do. Furthermore unlike in constraint databases where all the tuples that satisfy the constraints are in the database simultaneously, our models are continuous functions where given a time point there is only one location for the object.

An important body of work is done on indexing the moving object databases to efficiently answer various queries. [80] use PMR-Quadtrees [74] for indexing the trajectories extended some fixed number of time units in the future. The trajectories are one-dimensional and the index needs to be rebuilt periodically. The main drawback of the approach is extensive data replication. [47] transforms a line to a point and uses of regular spatial indexes such as Kd-trees.This method also works for only one dimensional movement. [7, 2] propose kinetic main-memory data structures for indexing mobile objects. The idea is to identify critical events and update the index only when those events occur, e.g., when two points pass each other. The index is not usable for queries related to time points before the last update point. Indexes using time-parameterized bounding rectangles include [65, 11].In this approach the indexed points and the bounding rectangles are augmented with velocity vectors thus they can be computed for different time points. Finally [73] extends this approach for expiring data.

In this work we proposed a simple yet effective indexing mechanism for the atoms of a go theory. We would like to note that it is a starting point but can be enhanced with the ideas from indexing moving object databases.

11.3 Collision Detection

In some respect our far-atoms detect the possible collision or proximity violation between two objects. Collision detection and proximity queries are studied in many previous work regarding robot navigation, mechanical engineering and geometric solid modeling. The three main approaches used in these work are :

- bounding volume hierarchies [16, 39, 8]
- swept volumes [12, 40]
- incremental distance computation [63, 64].

A rich survey of previous collision detection/proximity query algorithms can be found in [49]. To the best of our knowledge there is no previous work that addresses the kinds of uncertainties a go atom can represent. That is previous work on collision detection and proximity queries assume either the location of an object at a given time is known or in the case where there is no closed formula for the location of the object collision detection is performed only for sampled time points. Our work is different in many ways. The velocity of the object, the start and end times for the movement may not be known a priory. We have however an estimate interval for each of these variables. Using these estimates and the assumption of linear trajectories we were able to reduce the proximity queries under such uncertainties into spatio-temporal volume intersection problem with simple boundary equations.

11.4 Motion Planning

In some aspect deconfliction w.r.t. some integrity constraints can be seen as obstacle avoidance where each integrity constraint represents an obstacle in space-time. Although our deconfliction definition restricts us to existing paths we use modifications on the speed and time intervals to avoid or collide obstacles (for the case of goal constraints). Hence in space-time we are searching for a path from a set of points to another set of points that would avoid undesired regions in the related time interval. However for the goal constraints an object need to collide with the obstacle in a specific way (not every collision yields to entailment) so our paths not only avoids but sometimes seeks the obstacle. Immense amount of work has been done in the area of path planning, robot motion planning for obstacle avoidance. [36, 41, 13, 67] are excellent surveys for the algorithms and complexity of some problems, most general case of which is PSpace complete. However none of these algorithms are directly applicable to our case. For example the obstacles in our problems appear and disappear, we are not looking for a path between two points in space-time, instead due to spatio-temporal uncertainty encoded in the go-theories, we are searching for a path between a set of source points (possibly infinite) and a set of destination points (possibly infinite). Also the speed requirement on the objects complicates the problem even more. As discussed thoroughly in [26] kinodynamic motion planning is harder than non-constrained motion planning.

11.5 Default Reasoning

In a monotonic logic when new rules are added to a knowledge base, all the inferences still hold, thus the set of beliefs grow monotonically. In non-monotonic reasoning, the set of beliefs does not grow monotonically over time which means when new knowledge is gained, some previous conclusions may be retracted. For most of the reasoning tasks humans use non monotonic reasoning. That is we jump to conclusions and perform some default reasoning when there is not enough information.

In logics default reasoning can be formalized in one of two ways: (1) deal with arbitrary default assumptions as done in default logic and answer set programming (2) Formalize the specific default assumption that facts that are not known to be true can be assumed false by default as done in closed world assumption and circumscription.

Default logic is a non-monotonic logic proposed by Ray Reiter [69] to formalize reasoning with default assumptions. Default logic introduces a new inference rule which states if A is deducible and it is consistent to assume B then conclude C. This way of reasoning seems to be useful in explanation finding. For our purposes it is not possible to write default rules that would minimize the change in our interpretations w.r.t. time.

The Closed World Assumption (CWA) proposed by [68] holds that anything that cannot be entailed by a theory is false. Minker [55] extended the CWA to a Generalized CWA (GCWA) that accounts for disjunction. GCWA states that a formula is false if it is false in all minimal models of the theory. The go-theories proposed in this work are disjunctive because the start and end times and object velocities are all known to be within a given range. The notion of a coherent model of a go-theory selects certain models (much like Minker selected minimal models in GCWA) and uses these to make closed world inferences.

The only example of applying closed world assumption to moving objects is [42, 43]. They have used the CWA to track moving objects in football games using computer vision algorithms. They use CWA to adaptively select and weight image features used for correspondence. No motion reasoning of the type we perform in MCWA is done in this work.

Circumscription is a non-monotonic logic introduced by John McCarthy [53] It is a rule of conjecture that allows you to jump to the conclusion that the objects you can show that posses a certain property, p, are in fact all the objects that posses that property. This minimization is similar to the closed world assumption that what is not known to be true is false. Circumscription is not adequate for LOM predicates as our models refer to time and we are interested in minimization of changes in models with respect to time.

11.6 Resolving Inconsistencies in Logical Theories

There has been extensive work on resolving inconsistencies in logical theories — efforts include belief revision [27], knowledge-base revision [61] and answer set programming [58]. In all these efforts, an inconsistent theory is weakened to recover the consistency. In our deconfliction framework, the theory must *entail* the integrity constraints. Hence, we study methods to *strengthen* the theory so that it entails the integrity constraints. Consequently, our notion of deconfliction is quite different from this work on resolving inconsistencies.

Chapter 12

Conclusions

12.1 Summary

The thesis presents LOM for reasoning about flexible plans for moving objects. LOM is rooted in logic, constraint satisfaction and geometric reasoning. It is the first quantitative logic that explicitly represents the uncertainty about the start and end times of the movements as well as the speed of the movements. The thesis investigates the computational complexity of reasoning with such plans and provides efficient algorithms to check the consistency, i.e. realizability of these plans. It also identifies a tractable class of the problem which has a large number of applications.

Query answering with respect to flexible plans are also investigated throughly. The thesis introduces three different query atoms, namely: in, near and far which ask the possible locations of the object and its proximity to other objects respectively. The thesis provides very efficient algorithms to answer ground and non ground queries. The efficiency of the algorithms are verified by extensive empirical evaluation.

For plans that are temporally incomplete, i.e. does not specify what to do for some time intervals, a motion closed world assumption (MCWA) is proposed as a default reasoning mechanism. MCWArestricts the reasoning within a class of preferred models thus allows us to make more intelligent inferences.

Finally a theory for deconflicting plans for moving objects with respect to some integrity constraints is presented. The idea is to modify the existing theory so that any model of the theory would satisfy the integrity constraints. Based on this notion the *deconfliction* of a theory is defined. Finally the thesis presents an algorithm for efficiently computing a deconfliction.

12.2 Impact and Contributions

The contributions of this thesis are as follows:

- It provides a general framework for reasoning about plans for moving objects under uncertainty which is critical in many applications such as air traffic management and military mission planning. Using the declarative semantics it is also possible to customize this framework for different applications and couple it with other existing logics to create a more powerful theory.

- It describes the first quantitative logic to represent movements of objects while explicitly modeling temporal and speed uncertainty. Thus, it bridges the gap between geometric reasoning and spatio-temporal logics

- It presents theoretical results regarding the complexity of reasoning with flexible plans for moving objects and identifies a tractable class of the problem.

- It provides very efficient algorithms to reason about the flexible movement plans and an implementation that shows the effectiveness of the algorithms.

- It presents a default reasoning mechanism for incomplete theories thus allowing more intelligent inferences.

- It describes a theory and algorithms to deconflict plans for moving objects w.r.t. certain integrity constraints which is a critical task in many applications.

The declarative semantics of LOM makes it suitable for many applications and allows it to be open to various extensions such as the addition of new queries and customization for different applications. Moreover, the semantics of LOM lets us couple it with already existing logics, to create more expressive languages. For example, if we couple LOM with a description logic and spatial logic, we can express queries of the form, "Find me all the *commercial planes* flying over *France*.", where the concept of a commercial plane is given in the description logic and France is described in a spatial logic. Thus the theory presented in this thesis is a first step towards building a general-purpose plan management system for moving objects.

12.3 Future Work

This thesis provides the basic frame work for reasoning about plans for moving objects. There are several different research directions which can improve and extend this work.

- An obvious way to extend this work is to introduce additional atoms into the theory. Examples of such atoms would be (i) `wait` atom, which would enforce the object to wait a certain location and or a region through out an interval. (ii) Existential near and far atoms which would ask proximity queries for any point in a given time interval, rather than throughout the entire interval.

- In this work we estimated the routes in the go atoms to be line segments. While this is enough for most of the applications (any curve can be modeled as piecewise continuous line segments) for some applications the path can be as vague as a region. For these cases a more general path model is needed.

- The motion closed world assumption introduced in this work is one of many. Depending on the application the desired set of models can be different from the coherent models we investigated. For example another closed world assumption could limit the objects the objects to a certain region centered at their previous locations.

- The deconfliction problem investigated in this work is limited to goal regions and no-go regions. However there are many applications for which deconfliction with respect to proximity to other objects are important. This work does not address such problems.

- A major extension to LOM would be incorporating probabilistic reasoning so that we will be able to answer queries of the form: *With what probability is my ship going to encounter an enemy in this region?*.

- Finally an important future work will be generating flexible plans for moving objects in the context of LOM. Although several aspects of motion planning is studied in previous work, none of them are adequate to reason with the flexible plans that LOM can represent.

BIBLIOGRAPHY

[1] Amphibious forces. http://www.fas.org/man/dod-101/sys/ship/amphibious.htm.

[2] Pankaj K. Agarwal, Lars Arge, and Jeff Erickson. Indexing moving points. In *Symposium on Principles of Database Systems*, pages 175–186, 2000.

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

[4] Shyamanta M. Hazarika Anthony G. Cohn. Qualitative spatial representation and reasoning: An overview. *Fundam. Inform.*, 46(1-2):1–29, 2001.

[5] C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL*, 8(5):653–679, 2000.

[6] F. Wolter B. Konev, R. Kontchakov and M. Zakharyaschev. On dynamic topological and metric logics. *Studia Logica*, 2006. Accepted.

[7] Basch, Guibas, and Hershberger. Data structures for mobile data. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1997.

[8] Julien Basch, Jeff Erickson, Leonidas J. Guibas, John Hershberger, and Li Zhang. Kinetic collision detection between two simple polygons. *Comput. Geom. Theory Appl.*, 27(3):211–235, 2004.

[9] Marianne Baudinet, Marc Niézette, and Pierre Wolper. On the representation of infinite temporal data and queries (extended abstract). In *PODS '91: Proceedings*

of the tenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pages 280–290, New York, NY, USA, 1991. ACM Press.

[10] B. Bennett. Modal logics for qualitative spatial reasoning. *Journal of the Interest Group on Pure and Applied Logic*, 4:23–45, 1996.

[11] Mengchu Cai, Dinesh Keshwani, and Peter Z. Revesz. Parametric rectangles: A model for querying and animation of spatiotemporal databases. In Carlo Zaniolo, Peter C. Lockemann, Marc H. Scholl, and Torsten Grust, editors, *Advances in Database Technology - EDBT 2000, 7th International Conference on Extending Database Technology*, volume 1777 of *Lecture Notes in Computer Science*, pages 430–444. Springer, March 2000.

[12] S. Cameron. Collision detection by four-dimensional intersection testing. In *In Proc. IEEE Internat. Conf. Robot. Autom.*

[13] John F. Canny. *The complexity of robot motion planning*. MIT Press, Cambridge, MA, USA, 1988.

[14] J. Chomicki and P. Z. Revesz. Constraint-based interoperability of spatiotemporal databases. In *Proceedings of the 5th International Symposium on Large Spatial Databases (SSD)*, volume 1262, pages 142–161, 1997.

[15] Jan Chomicki and Tomasz Imieliński. Temporal deductive databases and infinite objects. In *PODS '88: Proceedings of the seventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 61–73, New York, NY, USA, 1988. ACM Press.

[16] J. D. Cohen, M. C. Lin, D. M., and M. K. Ponamgi. I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In *Symposium on Interactive 3D Graphics*, pages 189–196, 1995.

[17] A. G. Cohn and S. M Hazarika. Continuous transitions in mereotopology. In *Commonsense-2001: 5th Symposium on Logical Formalizations of Commonsense Reasoning*, 2001.

[18] 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, 2003.

[19] Anthony G Cohn and Nicholas Mark Gotts. Spatial regions with undetermined boundaries. In *Proceedings of Gaithesburg Workshop on GIS*. ACM, December 1994.

[20] Anthony G Cohn and Nicholas Mark Gotts. A theory of spatial regions with indeterminate boundaries. In C Eschenbach, C Habel, and B Smith, editors, *Topological Foundations of Cognitive Science*, 1994.

[21] Anthony G Cohn and Nicholas Mark Gotts. A mereological approach to representing spatial vagueness. In *Working Papers of the 9th International Workshop on Qualitative Reasoning*, pages 246–255, 1995.

[22] Anthony G Cohn and Nicholas Mark Gotts. The 'egg-yolk' representation of regions with indeterminate boundaries. In P Burrough and A M Frank, editors, *Proceedings,*

GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries, pages 171–187. Francis Taylor, 1996.

[23] Anthony G Cohn and Nicholas Mark Gotts. Representing spatial vagueness: a mereological approach. In J Doyle L C Aiello and S Shapiro, editors, *Proceedings of the 5th conference on principles of knowledge representation and reasoning (KR-96)*, pages 230–241. Morgan Kaufmann, 1996.

[24] A. Cohn D. Randell, Z. Cui. A spatial logic based on regions and connection. In *KR199: International Conference on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, 1992.

[25] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[26] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *J. ACM*, 40(5):1048–1066, 1993.

[27] Aldo Franco Dragoni. Belief revision: from theory to practice. *Knowl. Eng. Rev.*, 12(2):147–179, 1997.

[28] Max J. Egenhofer. Reasoning about binary topological relations. In Oliver Günther and Hans-Jörg Schek, editors, *Advances in Spatial Databases, Second International Symposium, SSD'91, Zürich, Switzerland, August 28-30, 1991, Proceedings*, volume 525 of *Lecture Notes in Computer Science*, pages 143–160. Springer, 1991.

[29] Robert Ellis and Denny Gulick. *Calculus with Analytic Geometry*. Harcourt Brace Jovanovich, New York, 1978.

[30] 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.

[31] Luca Forlizzi, Ralf Hartmut Güting, Enrico Nardelli, and Markus Schneider. A data model and data structures for moving objects databases. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 319–330, 2000.

[32] David Gabelaia, Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. 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.

[33] David Gabelaia, Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. Combining spatial and temporal logics: expressiveness vs. complexity. *Journal of Artificial Intelligence Research (JAIR)*, 23:167–243, 2005.

[34] M. Garey and D. Johnson. Two-processor scheduling with start-time and deadlines. *SIAM J. Cornput*, 6:416–426, 1977.

[35] Kurt Gödel. Eine interpretation des intuitionistischen aussagenkalküls. In *Ergebnisse eines mathematischen Kolloquiums*, volume 4, pages 34–38. 1933.

[36] J. E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*, chapter Motion planning by M. Sharir, pages 733–754. CRC Press, Florida, 1997.

[37] N M Gotts. An axiomatic approach to topology for spatial information systems. Technical report, Report 96.25, School of Computer Studies, University of Leeds, 1996.

[38] R. Güting, M. Böhlen, M. Erwig, C. Jensen, N. A. Lorentzos, L. Schneider, and M. Vazirgianis. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, 25(1), 2000.

[39] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, 15(3):179–210, 1996.

[40] Philip M. Hubbard. Collision detection for interactive graphics applications. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):218–230, 1995.

[41] Yong K. Hwang and Narendra Ahuja. Gross motion planning: a survey. *ACM Comput. Surv.*, 24(3):219–291, 1992.

[42] S. Intille. Tracking using a local closed-world assumption: Tracking in the football domain, 1994. Master's Thesis, M.I.T. Media Lab.

[43] S. Intille, J. Davis, and A. Bobick. Real-time closed-world tracking. In IEEE CVPR, pages 697–703, 1997.

[44] Paris C. Kanellakis. Constraint programming and database languages: A tutorial. In *PODS*, pages 46–53, 1995.

[45] Narendra Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4(8):373–395, 1984.

[46] L. G. Khachiyan. A polynomial time algorithm for linear programming. *Soviet Math. Dokl.*, 20(1):191–194, 1979.

[47] George Kollios, Dimitrios Gunopulos, and Vassilis J. Tsotras. On indexing mobile objects. In *Symposium on Principles of Database Systems*, pages 261–272, 1999.

[48] Oliver Kutz, Holger Sturm, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev. Logics of metric spaces. *ACM Transactions on Computational Logic (TOCL)*, 4(2):260–294, 2003.

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

[50] J.W. Lloyd. *Foundations of logic programming*. Springer-Verlag, 1987.

[51] V. J. Lumelsky. On fast computation of distance between line segments. *Inform. Processing Letters*, 21:55–61, 1985.

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

[53] John L. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artif. Intell.*, 13(1-2):27–39, 1980.

[54] 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.

[55] J. Minker. On indefinite databases and the closed world assumpion. In *Proceedings of the* 6^{th} *Conference on Automated Deduction*, volume 138 of *Lecture Notes in Computer Science*, pages 292–308. Springer Verlag, 1982.

[56] 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.

[57] Philippe Muller. Space-time as a primitive for space and motion. In *FOIS'98*, pages 63–76, Amsterdam, 1998.

[58] Pascal Nicolas, Laurent Garcia, and Igor Stéphan. A possibilistic approach to restore consistency in answer set programming. In *NMR*, pages 306–312, 2004.

[59] Werner Nutt. On the translation of qualitative spatial reasoning problems into modal logics. In *KI '99: Proceedings of the 23rd Annual German Conference on Artificial Intelligence*, pages 113–124, London, UK, 1999. Springer-Verlag.

[60] Ivan E. Orlov. The calculus of compatibility of propositions (in Russian). *Matematicheskii Sbornik*, 35:263–286, 1928.

[61] Odile Papini. Knowledge-base revision. *Knowl. Eng. Rev.*, 15(4):339–370, 2000.

[62] Jan Paredaens, Jan Van den Bussche, and Dirk Van Gucht. Towards a theory of spatial database queries (extended abstract). In *PODS '94: Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 279–288, New York, NY, USA, 1994. ACM Press.

[63] M. K. Ponamgi, D. Manocha, and M. C. Lin. Incremental algorithms for collision detection between polygonal models:. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):51–64, /1997.

[64] Madhav Ponamgi, Dinesh Manocha, and Ming C. Lin. Incremental algorithms for collision detection between solid models. In *SMA '95: Proceedings of the third ACM symposium on Solid modeling and applications*, pages 293–304. ACM Press, 1995.

[65] Cecilia Magdalena Procopiuc, Pankaj K. Agarwal, and Sariel Har-Peled. Star-tree: An efficient self-adjusting index for moving objects. In *ALENEX '02: Revised Papers from the 4th International Workshop on Algorithm Engineering and Experiments*, pages 178–193, London, UK, 2002. Springer-Verlag.

[66] 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.

[67] John Reif and Micha Sharir. Motion planning in the presence of moving obstacles. *J. ACM*, 41(4):764–790, 1994.

[68] Raymond Reiter. On closed world data bases. In Hervé Gallaire and Jack Minker, editors, *Logic and Data Bases*, pages 55–76, 1977.

[69] Raymond Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980.

[70] Jochen Renz and Bernhard Nebel. On the complexity of qualitative spatial reasoning: a maximal tractable fragment of the region connection calculus. *Artificial Intelligence*, 108(1-2):69–123, 1999.

[71] Mark Reynolds. The complexity of temporal logic over the reals. *Submitted to Elsevier Science*, 1999.

[72] Mark Reynolds. The complexity of the temporal logic with "until" over general linear time. *Journal of Computer and System Sciences*, 66(2):393–426, 2003.

[73] Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, and Mario A. Lopez. Indexing the positions of continuously moving objects. In *SIGMOD Conference*, pages 331–342, 2000.

[74] Hanan Samet. *The design and analysis of spatial data structures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

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

[76] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

[77] A. Prasad Sistla, Ouri Wolfson, Sam Chamberlain, and Son Dao. Modeling and querying moving objects. In *ICDE*, pages 422–432, 1997.

[78] John G. Stell and Michael F. Worboys. The algebraic structure of sets of regions. In *COSIT '97: Proceedings of the International Conference on Spatial Information Theory*, pages 163–174, London, UK, 1997. Springer-Verlag.

[79] J. Su, H. Xu, and O. H. Ibarra. Moving objects: Logical relationships and queries. In *Proceedings of the Seventh International Symposium on Spatial and Temporal Databases*, pages 3–19, 2001.

[80] J. Tayeb, O. Ulusoy, and O. Wolfson. A quadtree based dynamic attribute indexing method. *The Computer Journal*, 41(3):185–200, 1998.

[81] Goce Trajcevski, Ouri Wolfson1, Fengli Zhang, and Sam Chamberlain. Managing uncertainty in moving objects databases. In *ACM Transactions on Database Systems (TODS)*, pages 463–507, 2004.

[82] Ouri Wolfson, Sam Chamberlain, Son Dao, Liqin Jiang, and Gisela Mendez. Cost and imprecision in modeling the position of moving objects. In *ICDE*, pages 588–596, 1998.

[83] Ouri Wolfson, Bo Xu, Sam Chamberlain, and Liqin Jiang. Moving objects databases: Issues and solutions. In *Statistical and Scientific Database Management*, pages 111–122, 1998.

[84] Frank Wolter. Properties of tense logics. *Mathematical Logic Quarterly*, 42:481–500, 1996.

[85] Frank Wolter and Michael Zakharyaschev. Spatial reasoning in rcc-8 with boolean region terms. In *ECAI2000: Principles of Knowledge Representation and Reasoning*, pages 244–248, Berlin, 2000. IOS Press.

[86] Frank Wolter and Michael Zakharyaschev. Spatial reasoning in rcc-8 with boolean region terms. In W. Horn, editor, *Proceedings of the fourteenth European Conference on Artificial Intelligence, ECAI 2000*, pages 244–248. IOS Press, 2000.

[87] Frank Wolter and Michael Zakharyaschev. 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.

[88] Frank Wolter and Michael Zakharyaschev. Qualitative spatiotemporal representation and reasoning: a computational perspective. *Exploring artificial intelligence in the new millennium*, pages 175–215, 2003.

[89] Fusun Yaman, Dana Nau, and V S Subrahmanian. A logic of motion. In *Proceedings of KR2004*, pages 85–94, 2004.

Appendix A

Proofs

This appendix gives the proofs of the theorems and the lemmas in the main text. Lemmas and corollaries that have a label starting with a letter, does not appear in the text. They are used for proving the complex theorems presented in the text.

Consistency Checking

Lemma 1 *If two go atoms g_1, g_2 are **compatible** then*

1. *Direction of g_1 and g_2 are same, i.e $\vec{v}(g_1) = \vec{v}(g_2)$,*
2. *The intersection of $LS(g_1)$ and $LS(g_2)$ is a line segment,*
3. *The atoms temporally overlap, i.e. $t_1^-(g_1) \leq t_2^+(g_2)$ and $t_1^-(g_2) \leq t_2^+(g_1)$*
4. *The speed intervals overlap, i.e. $v^-(g_1) \leq v^+(g_2)$ and $v^-(g_2) \leq v^+(g_1)$*

Proof If g_1 and g_2 are compatible then by definition of compatibility (Definition 7) there is an interpretation $\mathcal{I}$ and there are time intervals $T = [t_1, t_2]$ and $T' = [t'_1, t'_2]$ such that $\mathcal{I}$ satisfies g over T and g' over T'. Furthermore, by the same definition, $t_2 > t'_1$ and $t'_2 > t_1$. Thus there is a time interval $A = [a, b]$ such that $A = T \cap T'$ and $a < b$. This means that A can not contain a single time instant. Then the following are trivial by Definition 9 (satisfaction of a movement w.r.t. an interval):

1. During T the object has to move in the direction of $\vec{v}(g_1)$ and similarly during T'

it has to move along $\vec{v}(g_2)$. Hence during A it has to move both in the direction of $\vec{v}(g_1)$ and $\vec{v}(g_2)$, which is possible only when the two directions are same.

2. For all time points in A the object can only be at points that are in the intersection of $LS(g_1)$ and $LS(g_2)$. The intersection can not be a single point P (i.e. end points of $LS(g_1)$ and $LS(g_2)$ meet) because that would imply the object stays at P during A which contradicts the satisfaction definition. Thus $LS(g_1) \cap LS(g_2)$ is a line segment.

3. By definition of satisfaction of a go atom over a time interval (Definition 5): $t_1^-(g_1) \leq t_1 \leq t_1^+(g_1)$ and $t_2^-(g_2) \leq t_2' \leq t_2^+(g_2)$ and by definition of compatibility $t_1 \leq t_2'$ hence $t_1^-(g_1) \leq t_1 \leq t_2' \leq t_2^+(g_2)$. We can show that $t_1^-(g_2) \leq t_2^+(g_1)$ holds similarly.

4. Again by Definition 5; $v^-(g_1) \leq d\mathcal{I}/dt \leq v^+(g_1)$ and $v^-(g_2) \leq d\mathcal{I}/dt \leq v^+(g_2)$ during A. Hence $v^-(g_1) \leq d\mathcal{I}/dt \leq v^+(g_2)$ and $v^-(g_2) \leq d\mathcal{I}/dt \leq v^+(g_1)$.

□

Lemma A-1 *Let G be a go theory and Γ be a concurrency graph of G. Let γ be movement of Γ. Suppose $g_1 \ldots g_n$ are the go atoms in γ then for all g_i, g_j $\vec{v}(g_i) = \vec{v}(g_j)$.*

Proof Assume there are g and g' such that $\vec{v}(g) \neq \vec{v}(g')$. Let $S(g)$ be the set of all g_i such that $\vec{v}(g_i) = \vec{v}(g)$. Let $S(g')$ be the set of all g_i such that $\vec{v}(g_i) = \vec{v}(g')$. Clearly $S(g)$ and $S(g')$ are disjoint. It follows from Lemma 1 that every edge (g_i, g_j) in Γ satisfy the compatibility conditions. By the same lemma if (g_i, g_j) is an edge in Γ and $g_i \in S(g)$ then $g_j \in S(g)$. Thus there can not be a path from g to g'. This is a contradiction because γ is a connected component of Γ. □

Lemma A-2 *Let G be a go theory and* Γ *be a concurrency graph of G. Let* γ *be movement of* Γ*. Suppose* $g_1 \ldots g_n$ *are the go atoms in* γ *then*

(1) $LS(g_1) \ldots LS(g_n)$ *are collinear; i.e. on the same infinite line.*

(2) $LS(g_1) \cup \ldots LS(g_n)$ *is a line segment.*

Proof (1) Assume there are g and g' such that $LS(g)$ and $LS(g')$ are not collinear. Let $S(g)$ be the set of all g_i such that $LS(g)$ and $LS(g_i)$ are collinear. Similarly let $S(g')$ be the set of all g_i such that $LS(g')$ and $LS(g_i)$ are collinear. Clearly $S(g)$ and $S(g')$ are disjoint. It follows from Lemma 1 that every edge (g_i, g_j) in Γ satisfy the compatibility conditions. Furthermore $LS(g_i)$ and $LS(g_j)$ are collinear because they share a line segment. Hence, when (g_i, g_j) is an edge in Γ and $g_i \in S(g)$ then $g_j \in S(g)$. Thus there can not be a path from g to g'. This is a contradiction because γ is a connected component of Γ.

Next we are going to show that (2) holds. Let $P_1 + (P_2 - P_1) * s$ be the parametric equation of a line L where $P_1 = loc_1(g^*)$ and $P_2 = loc_2(g^*)$ for an arbitrary $g^* \in G$. By (1), $LS(g) \subset L$ for all $g \in \gamma$. Furthermore for any point Q on L, there is a unique s_q such that $Q = P_1 + (P_2 - P_1) * s_q$.

Let $\{p_1, \ldots, p_k\} = \{loc_1(g_i), loc_2(g_i)\}_{i=1}^{n}$. Without loss of generality, assume that for any p_i, p_j; $i < j$ implies $s_{p_i} < s_{p_j}$.

Assume (2) is wrong, then there is a line segment $[p_i, p_{i+1}]$ such that $1 \leq i < k$ and $\forall g \in \gamma\ LS(g) \cap [p_i, p_{i+1}] = \emptyset$. Then for every $g \in \gamma$ either of the following holds:

(i) $loc_2(g) = p_j$ for some $j \leq i$

(ii) $loc_1(g) = p_j$ for some $j > i$

Then there are 3 cases to examine:

(Case 1) For every g, condition (i) holds. Then $p_{i+1}$is not an end point for any go atom in γ which is a contradiction.

(Case 2) For every g, condition (ii) holds. Then $p_i$is not a start point for any go atom in γ which is a contradiction.

(Case 3) Let S_a be the set of atoms that satisfy condition (i) and S_b be the set of atoms that satisfy (ii). Clearly $S_a \cup S_b$ contains all the atoms in γ and $S_a \cap S_b = \emptyset$. For any atom $g \in S_a$ and $g' \in S_b$; $LS(g) \cap LS(g') = \emptyset$ because all atoms in S_a is before p_i and all atoms in S_b is after p_{i+1}. By definition of compatibility conditions and concurrency graph any edge (g,g') in Γ such that $g \in S_a$ implies that $g' \in S_a$ as well. Hence there can not be a path form any atom in S_a and to any atom in S_b contradicting the assumption that γ is a connected component of Γ.

$\square$

Lemma A-3 *Let G be a go theory and* Γ *be a concurrency graph of G. Let* γ *be movement of* Γ. *If* $\mathcal{I}$ *satisfies* γ *over* T *then*

1. $\forall t \in T, \mathcal{I}(o,t) \in LS(\gamma)$
2. $\forall t,t' \in T \mid t < t' \implies \mathsf{dist}(loc_1(\gamma), I(o,t)) < \mathsf{dist}(loc_1(\gamma), I(o,t'))$
3. $\forall g,g' \in \gamma$ *such that* g *and* g' *are compatible* $\mathcal{I}$ *satisfies them concurrently.*

Proof

1. It follows from definition 9 that for every time point $t \in T$ there is a go atom $g \in \gamma$ such that $\mathcal{I} \models^{T^g} g$ and $t \in T^g$. Also by Definition 5, for every $t \in T^g$, $\mathcal{I}(o,t) \in LS(g)$. Thus $\forall t \in T, \mathcal{I}(o,t) \in LS(\gamma)$.

2. It follows from definition 9 that for every time point $t \in T$ there is a go atom $g \in \gamma$ such that $\mathcal{I} \models^{T^g} g$ and $t \in T^g$. Also by Definition 5, $\forall t, t' \in T^g \mid t < t' \implies \mathsf{dist}(loc_1(g), I(o,t)) < \mathsf{dist}(loc_1(g), I(o,t'))$. Since all atoms are collinear and move in the same direction $\mathcal{I}(o,t)$ moves away from $loc_1(\gamma)$ at evey time point in T.

3. Assume g and g' are compatible and $\mathcal{I}$ does not satisfy them concurrently. Then $\mathcal{I}$ has to satisfy them one after another, assume g before g'. By Lemma 1, $LS(g) \cap LS(g')$ is a line segment. Hence after finishing g, i.e. arriving to $loc_2(g)$, object has to move in the reverse direction to reach $loc_1(g')$. But this contradicts (2) which says at all time points the object moves in one direction, i.e. away from the start point of the movement. Thus all compatible atoms in γ are satisfied concurrently, whether or not there is an edge in Γ.

□

Corollary A-1 *Let G be a go theory and Γ be a concurrency graph of G. Let γ be movement of Γ. If $\mathcal{I}$ satisfies γ over T then for every $g \in \gamma$ there is a unique interval $T^g \subset T$ such that $\mathcal{I}$ satisfies g over T^g. $[t_1, t_2]$ where $t_1, t_2 \in T$, $\mathcal{I}(o,t_1) = loc_1(g)$, $\mathcal{I}(o,t_2) = loc_2(g)$.*

Proof Since $\mathcal{I}$ satisfies γ over T by definition there is a time interval $T^g = [T_1, T_2]$ over which $\mathcal{I}$ satisfies g. Again by definition $\mathcal{I}(o,T_1) = loc_1(g)$ and $\mathcal{I}(o,T_2) = loc_2(g)$. It follows from lemma A-3 that $\mathcal{I}$ is a monotonic function during T so $\forall t, t' \in T\mathcal{I}(o,t) = \mathcal{I}(o,t') \implies t = t'$ hence there can not be another time point $t_1 \in T$ such that $T_1 \neq t_1$ and $\mathcal{I}(o,t_1) = loc_1(g)$. Thus there can not be another interval over which g is satisfied.

□

Lemma 2 *Let G be a go theory and Γ be a concurrency graph of G. Let γ be movement of Γ. Then (i) γ is coherent iff $\mathcal{L}(\gamma)$ has a solution (ii) Γ is coherent iff for all movement γ of Γ, $\mathcal{L}(\gamma)$ is solvable.*

Proof We are going to prove (i)-the first part of the lemma as (ii) trivially follows from (i).

If part: First we will show that if there is a solution of $\mathcal{L}(\gamma)$ then there is a time interval T and an interpretation $\mathcal{I}$ such that $\mathcal{I}$ satisfies γ over T.

Let $g_1 \dots g_n$ be the atoms in γ. Let $p_1, p_2 \dots p_k$ be the check points of γ and $T_1 \dots T_k$ be the variables associated with check points in $\mathcal{L}(\gamma)$. Let $t_1 \dots t_k$ be reals such that when every T_i is replaced with t_i in $\mathcal{L}(\gamma)$, all constraints are satisfied. Given $t_1 \dots t_k$ and $\mathcal{L}(\gamma)$ we construct a LOM interpretation $\mathcal{I}$ as follows:

- $\forall i, 1 \leq i < k$ and $\forall t \in [t_i, t_{i+1}]$ $\mathcal{I}(o,t) = p_i + (t - t_i) * v_i * \vec{(\gamma)}$
- $\forall t < t_1, \mathcal{I}(o,t) = p_1$
- $\forall t > t_k, \mathcal{I}(o,t) = p_k$

where $v_i = \mathsf{dist}(p_i, p_{i+1})/(t_{i+1} - ti)$.

Basically o moves along the straight line from p_1 to p_k with piecewise constant speed in between check points during the time interval $[t_1, t_k]$. Before and after the interval it stays put at the origin and destination of γ.

Let $TI_1 \dots TI_n$ be time intervals such that $TI_i = [t_j, t_h]$ and $loc_1(g_i) = p_j$ and $loc_2(g_i) = p_h$ for some $j < h$. The existence of such intervals follow from the definitions of check points and $\mathcal{L}(\gamma)$. Next we are going to show that $\mathcal{I} \models g$ over $[t_1, t_k]$. By definition 9, it is enough to show the following three items:

1 $\mathcal{I}$ satisfies $g_i \in \gamma$ over $TI_i = [t_j, t_h]$ because

- $t_1^-(g_i) \leq t_j \leq t_1^+(g_i)$ by definition of $\mathcal{L}(\gamma)$ and t_j is a solution for T_j.
- $t_2^-(g_i) \leq t_h \leq t_2^+(g_i)$ for the same reason as above.
- By construction of $\mathcal{I}$ $\forall t \in [t_j, t_h]$ $\mathcal{I}(o, t) \in LS(g_i)$ and o is moving in the direction g_i.
- By construction of $\mathcal{I}$ the speed of o is $v_j, \ldots v_{h-1}$ during the intervals $[t_j, t_{j+1}]$ $\ldots [t_{h-1}, t_h]$ respectively. By the constraints in $\mathcal{L}(\gamma)$, $v^-(g_i) \leq v_l \leq v^+(g_i)$ for all $j \leq l < h$.

2 If g_i and g_j are movements of γ and (g_i, g_j) is an edge of Γ then $TI_i \cap TI_j \neq \emptyset$.

Let $TI_i = [t_x, t_y]$ and $TI_j = [t_w, t_z]$. We need to show that $[t_x, t_y] \cap [t_w, t_z] \neq \emptyset$. By definition of compatibility conditions $L = LS(g_i) \cap LS(g_j)$ is a line segment. Furthermore $L = [p_a, p_b]$ for some $a < b$ such that $x \leq a \leq b \leq y$ and $w \leq a \leq b \leq z$. Clearly $[t_x, t_y] \cap [t_w, t_z] = [t_a, t_b]$.

3 $[t_1, t_k] = TI_1, \cup \cdots \cup TI_n$.

By definition of TI_i every $TI_i \subseteq [t_1, t_k]$ hence $TI_1, \cup \cdots \cup TI_n \subseteq [t_1, t_k]$. For equality we need to show that for all $[t_i, t_i + 1]$ there is a TI_j such that $[t_i, t_i + 1] \subseteq TI_j$. By Lemma A-2, for every $[p_i, p_{i+1}]$ there is a $g_j \in \gamma$ such that $[p_i, p_{i+1}] \subseteq LS(g_j)$. Hence for every $[t_i, t_{i+1}]$ there is a TI_j such that $[t_i, t_{i+1}] \subseteq TI_j$.

1,2 and 3 proves that $\mathcal{I}$ satisfies γ over $[t_1, t_k]$.

Only if part: We will show that if there is a time interval T and an interpretation $\mathcal{I}$ such that $\mathcal{I}$ satisfies γ over T then there is a solution of $\mathcal{L}(\gamma)$. Let $g_1 \dots g_n$ be the atoms in γ.

Let $p_1 \dots p_k$ be the check points of γ. As shown in Lemma A-3 $\mathcal{I}$ is a monotonic function over T hence there are unique time points $t_1 < t_2 \dots < t_k$ such that $t_i \in T$ and $\mathcal{I}(o, t_i) = p_i$. We are going to show that if we replace each T_i in $\mathcal{L}(\gamma)$ with t_i every constraint in $\mathcal{L}(\gamma)$ is satisfied.

Assume there is a constraint C in $\mathcal{L}(\gamma)$ that is not satisfied then C is one of the 4 types of constraints:

(C1) $t_1^-(g_j) \leq T_i \leq t_1^+(g_j)$ for some i, j such that $loc_1(g_j) = p_i$;

This is not possible because as shown in Corollary A-1 $\mathcal{I}$ satisfies g_j over a unique interval $[t_i, t_x]$ which by definition of satisfaction of a go atom $t_1^-(g_j) \leq t_i \leq t_1^+(g_j)$ holds.

(C2) $t_2^-(g_j) \leq T_i \leq t_2^+(g_j)$ for some i, j such that $p_i = loc_2(g_j)$;

This is not possible because as shown in Corollary A-1, $\mathcal{I}$ satisfies g_j over a unique interval $[t_x, t_i]$ which by definition of satisfaction of a go atom $t_2^-(g_j) \leq t_i \leq t_2^+(g_j)$ holds.

(C3) $dist(p_i, p_{i+1}) \leq (T_{i+1} - T_i) \times v_i^+$, for some $i = 1, \dots, k-1$; where $v_i^+ = \max\{v^+(g) \mid [p_i, p_{i+1}]$ is a subsegment of the line segment $[loc_1(g), loc_2(g)]\}$.

So this constraint is equivalent to:

$\forall g \in \gamma | [p_i, p_{i+1}] \subseteq LS(g)\ dist(p_i, p_{i+1}) \leq (T_{i+1} - T_i) \times v_i^+(g)$.

If C s violated then there is a $g \in \gamma$ such that $[p_i, p_{i+1}] \subseteq LS(g)$ and $dist(p_i, p_{i+1}) >$

$(t_{i+1} - t_i) \times v_i^+(g)$. It follows from Lemma A-3 and Corollary A-1 that $\mathcal{I}$ satisfies g over a unique interval T such that $[t_i, t_{i+1}] \subseteq T$. By definition of satisfaction over a time interval (Definition 5), for all $t \in T$, $v^-(g) \leq \frac{d\mathcal{I}}{dt} \leq v^+(g)$. Hence between any two time points in T the average speed of the object is less than or equal to $v^+(g)$. Thus there can not be any g that violates constraint $dist(p_i, p_{i+1}) \leq (T_{i+1} - T_i) \times v^+(g)$ proving that C must hold.

C4 $(T_{i+1} - T_i) \times v_i^- \leq dist(p_i, p_{i+1})$ for some $i = 1, \ldots, k-1$ where

$v_i^- = \max\{v^-(g) \mid [p_i, p_{i+1}]$ is a subsegment of the line segment $LS(g)$.

This leads to a contradiction and the proof is similar to (C3).

Thus $t_1 \ldots t_k$ is a solution to $\mathcal{L}(\gamma)$. $\square$

Theorem 2 *A go theory G is consistent iff there is a plan π for G.*

Proof The theorem has two parts.

If part: We are going to prove that if G is consistent then there is a plan $\pi = \langle \Gamma, \sqsubseteq \rangle$ for G.

We are going to show that for any interpretation $\mathcal{I} \models G$ there is a concurrency graph $\Gamma_{\mathcal{I}}$ of G and a total order $\sqsubseteq_{\mathcal{I}}$ compatible with $\Gamma_{\mathcal{I}}$ such that $\mathcal{C}(G, \Gamma_{\mathcal{I}}, \sqsubseteq_{\mathcal{I}})$ has a solution.

Let $g_1 \ldots g_k$ be the atoms in G and $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models G$. Then by definition of satisfaction of a go theory there are time intervals $TI_{g_1} \ldots TI_{g_k}$ such that $\mathcal{I} \models g_i$ over TI_{g_i}. Suppose for every $g \in G$ TI_g is of the form $[TI_{g1} TI_{g2}]$.

- Let $\Gamma_{\mathcal{I}}$ be a graph such that the nodes are the atoms in G and (g, g') is an edge in $\Gamma_{\mathcal{I}}$ iff $TI_g \cap TI_{g'} \neq \emptyset$. Note that by definition of compatibility any g, g' such that

$\mathcal{I}$ satisfies concurrently are compatible. $\Gamma_{\mathcal{I}}$ is a concurrency graph of G because every edge is between two compatible atoms.

- Let $\gamma_1 \ldots \gamma_n$ be the movements in $\Gamma_{\mathcal{I}}$. Let $\preceq$ be a partial order on movements of $\Gamma_{\mathcal{I}}$ such that $\gamma_i \preceq \gamma_j$ iff $\forall g \in \gamma_i\ \forall g' \in \gamma_j\ t_2^+(g) \leq t_1^-(g')$.

  Let $\sqsubseteq_{\mathcal{I}}$ be a total order on movements of $\Gamma_{\mathcal{I}}$ such that $\gamma_i \sqsubseteq_{\mathcal{I}} \gamma_j$ iff $\forall g \in \gamma_i \forall g \in \gamma_j\ TI_{g2} \leq TI_{g'1}$.

  By definition of satisfaction of a go atom for all g the following hold: $TI_{g1} \geq t_1^-(g)$ and $TI_{g2} \leq t_2^+(g)$.

  It follows $\gamma_i \preceq \gamma_j$ implies $\forall g \in \gamma_i \forall g' \in \gamma_j TI_{g2} \leq t_2^+(g) \leq t_1^-(g') \leq TI_{g'1}$.

  Hence $\gamma_i \preceq \gamma_j$ implies $\gamma_i \sqsubseteq_{\mathcal{I}} \gamma_j$. Thus $\sqsubseteq_{\mathcal{I}}$ is a topological sort of $\preceq$. Then by definition of compatible order $\sqsubseteq_{\mathcal{I}}$ is a total order compatible with $\Gamma_{\mathcal{I}}$.

Next step is to show that $\mathcal{C}(G, \Gamma_{\mathcal{I}}, \sqsubseteq_{\mathcal{I}})$ has a solution. Fist we will prove the following propositions:

(i) **Let γ be a movement of $\Gamma_{\mathcal{I}}$ then $T_\gamma = \bigcup_{g \in \gamma} TI_g$ is a single time interval.**

Assume T_γ is not a single time interval. Then there are three time points $t^- < t < t^+$ such that $t^- \in T_\gamma$, $t^+ \in T_\gamma$ and $t \notin T_\gamma$.Then $\forall g \in \gamma,\ t \notin TI_g$. Then atoms can be partitioned into two sets; containing atoms whose TI_g is before or after t respectively. There can be no edge between atoms in different partitions. Which means the two partitions are not connected. This is a contradiction because by definition γ is a connected component of $\Gamma_{\mathcal{I}}$

(ii) **$\mathcal{I}$ satisfies γ over the interval T_γ.**

$\mathcal{I}$ satisfies every $g \in \gamma$ over TI_g. By definition of $\Gamma_{\mathcal{I}}$, $\mathcal{I}$ concurrently satisfies every

g, g' such that (g, g') is an edge in $\Gamma_{\mathcal{I}}$. Finally by (i) $T_\gamma = \bigcup_{g \in \gamma} TI_g$.

Let γ be a movement of $\Gamma_{\mathcal{I}}$ such that $\mathcal{I}$ satisfies γ over T_γ. Now suppose that $Checkpoints(\gamma) = [p_1^\gamma, \ldots, p_z^\gamma]$ and $T_1^\gamma, T_2^\gamma \ldots T_z^\gamma$ are the variables in $\mathcal{L}(\gamma)$ such that $T_i^\gamma = Var(\gamma, P_i^\gamma)$. Suppose θ_γ maps each T_i^γ to t_i^γ such that $t_i^\gamma \in T_\gamma$ and $\mathcal{I}(o, t_i^\gamma)$=$p_i^\gamma$. By corollary A-1 such t_i^γ exists and unique. Then as shown in proof of Lemma 2, θ_γ is a solution for γ.

Next we will show that $\theta = \bigcup_{\gamma \in \Gamma_{\mathcal{I}}} \theta_\gamma$ **is a solution for** $\mathcal{C}(G, \Gamma_{\mathcal{I}}, \sqsubseteq_{\mathcal{I}})$. Clearly for any movement γ, θ is a solution to constraints in $\mathcal{L}(\gamma)$. So it is sufficient to show that θ satisfies the following constraints:

- for every movement pair γ, γ' of $\Gamma_{\mathcal{I}}$ such that $\gamma \sqsubseteq_{\mathcal{I}} \gamma'$ and $loc_2(\gamma) = loc_1(\gamma')$,

  $$Var(\gamma, P) \leq Var(\gamma', Q) \text{ where } P = loc_2(\gamma) \text{ and } Q = loc_1(\gamma');$$

  Assume this is violated for some pair of movements γ and γ' such that $\gamma \sqsubseteq_{\mathcal{I}} \gamma'$.

  By definition of θ every variable in $\mathcal{C}(G, \Gamma_{\mathcal{I}}, \sqsubseteq_{\mathcal{I}})$ related to a movement γ has a domain T_γ. If for some check point P of γ and some check point Q of γ', $\theta(Var(\gamma, P)) > \theta(Var(\gamma', Q))$ then $\gamma \not\sqsubseteq_{\mathcal{I}} \gamma'$ which is a contradiction.

- for every movement pair γ, γ' of $\Gamma_{\mathcal{I}}$ such that $\gamma \sqsubseteq_{\mathcal{I}} \gamma'$ and $loc_2(\gamma) \neq loc_1(\gamma')$,

  $$Var(\gamma, P) < Var(\gamma', Q) \text{ where } P = loc_2(\gamma) \text{ and } Q = loc_1(\gamma');$$

  Assume this is violated for some pair of movements γ and γ' such that $\gamma \sqsubseteq_{\mathcal{I}} \gamma'$. If for some check point P of γ and some check point Q of γ', $\theta(Var(\gamma, P)) > \theta(Var(\gamma', Q))$ then $\gamma \not\sqsubseteq_{\mathcal{I}} \gamma'$ which is a contradiction.

It follows from Lemma A-3 and definition of θ, $\theta(Var(\gamma, P)) = \theta(Var(\gamma', Q)) = t$ can only happen iff $P = loc_2(\gamma)$ and $Q = loc_1(\gamma')$. In this case, by definition of an interpretation $\mathcal{I}(o, t) = P$ and $\mathcal{I}(o, t) = Q$ hence $P = Q$ which is a contradiction for this type of constraint.

Only if part: If there is a plan $\pi = \langle \Gamma, \sqsubseteq \rangle$ for G then G is consistent.

By definition of a plan $\mathcal{C}(G, \Gamma, \sqsubseteq)$ has a solution. We are going to show that for any solution of $\mathcal{C}(G, \Gamma, \sqsubseteq)$ there is an interpretation $\mathcal{I} \models G$.

Let γ be a movement in Γ and $Checkpoints(\gamma) = p_1^\gamma, \ldots, p_k^\gamma$. Let $T_1^\gamma, T_2^\gamma \ldots T_k^\gamma$ be the variables in $\mathcal{L}(\gamma)$ such that $T_i^\gamma = Var(\gamma, P_i^\gamma)$. Let θ be a solution for $\mathcal{C}(G, \Gamma, \sqsubseteq)$. Then we build an interpretation $\mathcal{I}$ as follows:

- $\mathcal{I}(o, t) = p_i^\gamma + (t - t_i^\gamma) * v_i^\gamma * \vec{v}(\gamma)$, when $t_i^\gamma \leq t \leq t_{i+1}^\gamma$ and $1 \leq i < k$ where
  - γ is a movement in Γ
  - k is the total number of points in $Checkpoints(\gamma)$
  - p_i^γ is the i^{th} point in $Checkpoints(\gamma)$
  - t_i^γ is the value of variable T_i^γ in θ
  - $v_i^\gamma = \mathsf{dist}(p_i^\gamma, p_{i+1}^\gamma)/(t_{i+1}^\gamma - t_i^\gamma)$
- $\mathcal{I}(o, t)$ is any point such that $\mathcal{I}$ is continuous, otherwise.

Next we will to show that $\mathcal{I} \models G$. It suffices to show that for every movement γ , $\mathcal{I} \models \gamma$ over some interval. We are going to show this using the proof of Lemma 2. Let θ_γ be a subset of θ that contain only the mappings for the variables in $\mathcal{L}(\gamma)$. Then $\mathcal{I}$ is an interpretation that can be created in proof of Lemma 2 given the solution θ_γ. Then it follows from the first part proof of Lemma 2 that there is a time interval T such that $\mathcal{I} \models \gamma$ over T. $\square$

Theorem 3 *Algorithm* $\mathsf{Consistent}(G)$ *is correct, i.e.,* G *is consistent iff there is a way to make the nondeterministic choices in Step 2 and 4 such that the algorithm returns "yes."*

Proof First we will prove the statement "**If there is a way to make the nondeterministic choices such that the algorithm** $\mathsf{Consistent}(G)$ **returns "yes." then** G **is consistent.**"

In step 2 the algorithm nondeterministically creates a concurrency graph of G. In step 4 it nondeterministically imposes a total ordering $\sqsubseteq$ on the movements of Γ. If $\sqsubseteq$ is not compatible with Γ it returns false. The algorithm only returns true when the constraint set C has a solution. By theorem 2 when $C = \mathcal{C}(G, \Gamma, \sqsubseteq)$ has a solution, G is consistent.

Now we will prove the statement "**If** G **is consistent then there is a way to make the nondeterministic choices such that the algorithm** $\mathsf{Consistent}(G)$ **returns "yes.".**"

If G is consistent then there is a a plan $\langle \Gamma, \sqsubseteq \rangle$ for G. The nondeterministic steps 2 and 4 create every possible concurrency graph and the total order on the movements of the graph, hence one of the nondeterministic traces of the algorithm will build Γ and $\sqsubseteq$. $\square$

Simple Go-theories

Lemma 3 *A simple go theory G is consistent iff there exists a plan* $\pi = \langle \Gamma^*, \sqsubseteq \rangle$ *for G where* Γ^* *is the maximal concurrency graph of G.*

Proof If there exists a plan $\pi = \langle \Gamma^*, \sqsubseteq \rangle$ for G then G is consistent follows from Theorem 2.

By Theorem 2 we know that if G is consistent then there is a plan $\langle \Gamma, \sqsubseteq \rangle$ for G. Assume $\Gamma = \langle V, E \rangle$ and $\Gamma^* = \langle V, E^* \rangle$ are different. Since the $V = G$ for all concurrency graphs of G there are two cases to consider:

- There is a pair of atoms g, g' such that $(g, g') \in E$ and $(g, g') \notin E^*$: This is a contradiction because by definition of maximal graphs (Definition 18) Γ^* contains all possible edges.
- There is a pair of atoms g, g' such that $(g, g') \in E^*$ and $(g, g') \notin E$: In this case we need to investigate two cases again:
  - If Γ and Γ^* have the same movements then $\langle \Gamma^*, \sqsubseteq \rangle$ is also plan for G. This is because by definition 15, $\mathcal{C}(G, \Gamma, \sqsubseteq)$ and $\mathcal{C}(G, \Gamma*, \sqsubseteq)$ contain the same constraints.
  - If Γ and Γ^* do not have the same movements then w.r.t. plan $\langle \Gamma, \sqsubseteq \rangle$ either g or g' is satisfied before the other one. Thus by definition of necessary/maximal graphs $\Gamma*$ can not contain the edge (g, g') which leads to a contradiction.

$\square$

Theorem 4 *Let G be a simple go theory,* Γ^* *be the maximal concurrency graph of G and* $\sqsubseteq^*$ *be the ordering in Definition 20. G is consistent iff*

- Γ^* *is coherent, and*
- $\sqsubseteq^*$ *is a total order, and*
- $\mathcal{C}(\Gamma^*, \sqsubseteq^*)$ *has a solution*

Proof Once again the proof has two parts. We will first show that if the three conditions are satisfied G is consistent. For that we need to show the following:

- Γ^* is a concurrency graph of G. This is true by definition of maximal graphs.
- $\sqsubseteq^*$ is compatible with Γ^*. Remember that for compatibility $\sqsubseteq^*$ has to be a topological sort of the partial order $\preceq$ in Definition 14. Let γ and γ' be two movements of Γ^*. It is easy to see that whenever $\gamma \preceq \gamma'$, $\gamma \sqsubseteq^* \gamma'$ is also true.

Hence if all the conditions are true, it follows from the definition of a plan that $\langle \Gamma^*, \sqsubseteq^* \rangle$ is a plan for G. Finally by Theorem 2, G is consistent.

Next we are going to show that if G is consistent then the three conditions hold.

- By Lemma 3 if G is consistent then there is a plan $\langle \Gamma^*, \sqsubseteq \rangle$ for G. If Γ^* is incoherent then $\mathcal{C}(G, \Gamma^*, \sqsubseteq)$ has no solution which leads to a contradiction.
- Assume $\sqsubseteq^*$ is not a total order. Then there are two movements γ, γ' in Γ^* such that either of the following is true:
  - $\gamma \sqsubseteq^* \gamma'$ and $\gamma' \sqsubseteq^* \gamma$: If this is the case than by Definition 19, G can not be a simple go theory.
  - Neither $\gamma \sqsubseteq^* \gamma'$ nor $\gamma' \sqsubseteq^* \gamma$: In this case none of the two movements can finish before the other one starts. Hence there is no way to achieve both of them in an order so G has to be inconsistent which contradicts our initial assumption.
- From Lemma 3 we know that there is a plan $\langle \Gamma^*, \sqsubseteq \rangle$ for G. Assume $\sqsubseteq$ is different from $\sqsubseteq^*$. Then there are movements γ, γ' such that $\gamma \sqsubseteq \gamma'$ and $\gamma' \sqsubseteq^* \gamma$. Further-

more since $\langle \Gamma^*, \sqsubseteq \rangle$ is a plan, $\mathcal{C}(G, \Gamma^*, \sqsubseteq)$ has a solution. Thus by Definition 15 the constraint $\{Var(\gamma, P) < Var(\gamma', Q)$ where $P = loc_2(\gamma)$ and $Q = loc_1(\gamma')\}$ has a solution. This constraint ensures γ's end time is before the start time of γ'. Thus it is possible for γ to finish before γ' and vice versa because $\gamma' \sqsubseteq^* \gamma$. But this contradicts with the definition of a simple theory, hence we need to conclude that $\sqsubseteq = \sqsubseteq^*$. This demonstrates that $\mathcal{C}(G, \Gamma^*, \sqsubseteq^*)$ has a solution.

□

Theorem 5 *Suppose G is a consistent simple go theory. Then all plans for G are equal to the plan $\pi^* = \langle \Gamma^*, \sqsubseteq^* \rangle$ where Γ^* is the maximal concurrency graph of G and $\sqsubseteq^*$ is the total order given in definition 20.*

Proof Let $\pi = \langle \Gamma, \sqsubseteq \rangle$ be a plan for G such that π is not equivalent to π^* then either of the following holds:

- There is an $\mathcal{I}$ such that $\mathcal{I}$ is an instance of π and not an instance of π^*: This might be caused by one of the following:
  - π and π^* have different movements. This leads to a contradiction as shown in Lemma 3.
  - π and π^* have same movements but $\sqsubseteq$ and $\sqsubseteq^*$ are different. This means that there are movements γ, γ' such that $\gamma \sqsubseteq \gamma'$ and $\gamma \sqsubseteq^* \gamma'$. Thus it is possible for γ to finish before γ' and vice versa. But this contradicts with the definition of a simple theory, hence the case can not be true.

- There is an $\mathcal{I}$ such that $\mathcal{I}$ is an instance of π^* and not an instance of π: This is also not possible and the proof is similar to the previous case.

$\square$

Temporal,Positional and Speed Certainty Intervals

Lemma 4 *Let P be a point, G be a go theory, o be an object,π be a plan for G^o. Let γ be a movement in π such that $P \in LS(\gamma)$ and $CheckPoints(\gamma) = [p_1 \ldots p_n]$. Then $T^-(G^o, \pi, \gamma, P)$ satisfies the following:*

(A) ***If*** *$P = p_i$ for some i then $T^-(G^o, \Gamma, \sqsubseteq, \gamma, P)$ is the solution of the following linear programming problem :* ***minimize*** *$Var(\gamma, P)$* ***subject to*** *$\mathcal{C}(G^o, \Gamma, \sqsubseteq)$*

(B) ***If*** *the previous case does not apply and P is on line segment $[p_i, p_{i+1}]$ for some i then $T^-(G^o, \pi, \gamma, P)$ is the* ***maximum*** *of the following two items:*

(i) $T^-(G^o, \pi, \gamma, p_i) + dist(P, p_i)/v^+(\gamma, p_i, p_{i+1})$

(ii) $T^-(G^o, \pi, \gamma, p_{i+1}) - dist(P, p_{i+1})/v^-(\gamma, p_i, p_{i+1})$

Proof (A) follows from the definition of $T^-(G^o, \pi, \gamma, P)$ and the proof of Theorem 4.1.3 which shows that for every solution of $\mathcal{C}(G^o, \Gamma, \sqsubseteq)$ there is a model $\mathcal{I}$ satisfying G^o and for every $\mathcal{I} \models G^o$ there is a solution for $\mathcal{C}(G^o, \Gamma, \sqsubseteq)$.

(B) If P is not a check point of γ then there is no variable in $\mathcal{C}(G^o, \Gamma, \sqsubseteq)$ that represents o's arrival time to P. Then P is on the line segment $[p_i, p_{i+1}]$ for some consecutive check points of γ. First we will show that $T^-(G^o, \pi, \gamma, P)$ is greater than or equal to (i) and (ii). It follows from satisfaction of a go atom and satisfaction of a movement that for all interpretation $\mathcal{I}$ that satisfy γ w.r.t. π over a time interval T, $\exists t_i, t_{i+1}, t_p \in T$ such that $t_i < t_P < t_{i+1}$ and $I(o,t_i) = p_i$, $I(o,t_P) = P$ and $I(o,t_{i+1}) = p_{i+1}$. Furthermore as the proof of the Lemma 2 establishes the speed of the object between p_i and p_{i+1} is always within $v^-(\gamma, p_i, p_{i+1})$ and $v^+(\gamma, p_i, p_{i+1})$. Hence the average speed of the object between any two points on $[p_i, p_{i+1}]$, is always within $v^-(\gamma, p_i, p_{i+1})$ and $v^+(\gamma, p_i, p_{i+1})$. So the following inequalities hold:

$$\frac{\mathsf{dist}(p_i,P)}{t_P - t_i} \leq v^+(\gamma, p_i, p_{i+1}) \text{ and } \frac{\mathsf{dist}(p_{i+1},P)}{t_P - t_{i+1}} \geq v^-(\gamma, p_i, p_{i+1})$$

which are equal to:

$$t_i + \frac{\mathsf{dist}(p_i,P)}{v^+(\gamma,p_i,p_{i+1})} \leq t_P \quad \text{and} \quad t_{i+1} - \frac{\mathsf{dist}(p_{i+1},P)}{v^-(\gamma,p_i,p_{i+1})} \leq t_P$$

By definition of earliest arrival time, the following also are true: $T^-(G^o, \pi, \gamma, p_i) \leq t_i$ and $T^-(G^o, \pi, \gamma, p_{i+1}) \leq t_{i+1}$. Combining these with the inequalities above we get:

$$T^-(G^o, \pi, \gamma, p_i) + \frac{\mathsf{dist}(p_i,P)}{v^+(\gamma,p_i,p_{i+1})} \leq t_P \quad \text{and} \quad T^-(G^o, \pi, \gamma, p_{i+1}) - \frac{\mathsf{dist}(p_{i+1},P)}{v^-(\gamma,p_i,p_{i+1})} \leq t_P$$

So in any $\mathcal{I}$, t_P is grater than or equal to (i) and (ii). Hence by definition of earliest arrival time, $T^-(G^o, \pi, \gamma, P)$ is also grater than or equal to (i) and (ii).

Next we will show that there is an $\mathcal{I}$ for which t_P is equal to maximum of (i) and (ii) and $\mathcal{I}$ is an instance of π. For readability let's assume $T^-(G^o, \pi, \gamma, p_i) = T_i$, $T^-(G^o, \pi, \gamma, p_{i+1}) = T_{i+1}$, $v^-(\gamma, p_i, p_{i+1}) = v^-$ and $v^+(\gamma, p_i, p_{i+1}) = v^+$.

By definition of earliest arrival time we know that there are interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ such that

- $\mathcal{I}_1(o, T_i) = p_i, \mathcal{I}_1(o, t_{i+1}) = p_{i+1}$ and $\mathcal{I}_1$ satisfies γ w.r.t. π oven an interval including T_i and t_{i+1}.

- $\mathcal{I}_2(o, t_i) = p_i, \mathcal{I}_2(o, T_{i+1}) = p_{i+1}$ and $\mathcal{I}_2$ satisfies γ w.r.t. π oven an interval including t_i and T_{i+1}.

Using $\mathcal{I}_1$ and $\mathcal{I}_2$ we will define a LOMinterpretation $\mathcal{I}$ as follows:

$$\mathcal{I}(o,t) = \begin{cases} \mathcal{I}_1(o,t) & \text{if } t \leq T_i \\ p_i + v^+(t - T_i)\vec{\gamma} & \text{if } T_i < t < T_{i+1} \wedge t < t^* \\ p_{i+1} - v^-(T_{i+1} - t)\vec{\gamma} & \text{if } T_i < t < T_{i+1} \wedge t^* \leq t \\ \mathcal{I}_2(o,t) & \text{if } t \geq T_{i+1} \end{cases}$$

where $t^* \in \mathbf{R}$ such that $p_i + v^+(t^* - T_i)\vec{\gamma} = p_{i+1} - v^-(T_{i+1} - t^*)\vec{\gamma}$

The continuity of $\mathcal{I}$ is not trivial so we will demonstrate that $\mathcal{I}$ is continuous before we continue with the rest of the proof. For this we need to show $T_i \leq t^* \leq T_{i+1}$ which is eliminates the only possible source of discontinuity in $\mathcal{I}$.

1. $t_i \geq T_i$ and $t_{i+1} \geq T_{i+1}$ by definition of earliest arrival time
2. $\frac{\mathsf{dist}(p_i,p_{i+1})}{v^+} \leq t_{i+1} - T_i \leq \frac{\mathsf{dist}(p_i,p_{i+1})}{v^-}$ by definition of satisfaction of a movement and interpretation $\mathcal{I}_1$
3. $T_{i+1} - T_i \leq \frac{\mathsf{dist}(p_i,p_{i+1})}{v^-}$ by 1 and 2.
4. $\frac{\mathsf{dist}(p_i,p_{i+1})}{v^+} \leq T_{i+1} - t_i \leq \frac{\mathsf{dist}(p_i,p_{i+1})}{v^-}$ by definition of satisfaction of a movement and interpretation $\mathcal{I}_2$
5. $\frac{\mathsf{dist}(p_i,p_{i+1})}{v^+} \leq T_{i+1} - T_i$ by 1 and 4.
6. $t^* = \frac{T_i \times v^+ - T_{i+1} \times v^- + \mathsf{dist}(p_i,p_{i+1})}{v^+ - v^-}$ by definition of t^* and some simple algebra
7. $t^* \geq T_i$ by 3 and 6
8. $t^* \leq T_{i+1}$ by 5 and 6

Now that we have established the continuity of $\mathcal{I}$, we need to show that $\mathcal{I}$ is an instance of π. It is trivial to see that because of the way $\mathcal{I}$ is constructed, all movements before and after γ are still satisfied. The only question is then to show that γ is still satisfied. The construction of $\mathcal{I}$ trivially satisfies the second and third conditions in Definition 9 (satisfaction of a movement). The first condition that states every go atom should be satisfied needs to be worked on, because between T_i and T_{i+1}, $\mathcal{I}$ differs both from $\mathcal{I}_1$ and $\mathcal{I}_2$. Specifically we need to show that for all atoms $g \in \gamma$ such that $[p_i, p_{i+1}] \subset LS(g)$ the speed constraints are satisfied throughout $[T_i$,$T_{i+1}]$. By definition of max/min speed allowed (Definition 13) for every such g: $v^-(g) \leq v^-(\gamma, p_i, p_{i+1}) \leq v^+(\gamma, p_i, p_{i+1}) \leq v^+(g)$. Hence by construction of $\mathcal{I}$ throughout $[T_i$,$T_{i+1}]$ the speed of the object satisfies the speed limits in every atom $g \in \gamma$ such that $[p_i, p_{i+1}] \subset LS(g)$.

Finally we need to demonstrate that for any point P on $[p_i, p_{i+1}]$ if $I(o, t_p) = P$ and $T_i \leq t_p \leq T_{i+1}$ then $t_p = max(T_i + dist(P, p_i)/v^+, T_{i+1} - dist(P, p_{i+1})/v^-)$. Let $I(o, t^*) = p^*$ then:

9. $t^* = T_i + \frac{\mathsf{dist}(p_i, p^*)}{v^+} = T_{i+1} - \frac{\mathsf{dist}(p_{i+1} p^*)}{v^-}$; by definition of $\mathcal{I}$ and t^*.

10. $\mathsf{dist}(p_i, P) \leq \mathsf{dist}(p_i, p^*)$ implies $T_i + \frac{\mathsf{dist}(p_i, P)}{v^+} \leq t^*$ and $t_p = T_i + \frac{\mathsf{dist}(p_i, P)}{v^+}$; follows from 9 and the definition of $\mathcal{I}$.

11. $\mathsf{dist}(p_i, P) \leq \mathsf{dist}(p_i, p^*)$ implies $T_{i+1} - \frac{\mathsf{dist}(p_{i+1}, P)}{v^-} \leq T_i + \frac{\mathsf{dist}(p_i, P)}{v^+}$

    When $P = p_i$, (11) follows from (3). Since both $T_{i+1} - \frac{\mathsf{dist}(p_{i+1}, P)}{v^-}$ and $T_i + \frac{\mathsf{dist}(p_i, P)}{v^+}$ are monotonically increasing linear functions which have the same value when $P = p^*$ we can also conclude that (11) is true for any point on $[p_i, p^*]$.

12. $\mathsf{dist}(p_i, P) \geq \mathsf{dist}(p_i, p^*)$ implies $T_{i+1} - \frac{\mathsf{dist}(p_{i+1}, P)}{v^-} \geq T_i + \frac{\mathsf{dist}(p_i, P)}{v^+}$

It follows from a similar reasoning that leads to (11).

13. $t_p = max(T_{i+1} - \frac{\mathsf{dist}(p_{i+1},P)}{v^-} \leq T_i + \frac{\mathsf{dist}(p_i,P)}{v^+})$ follows from (11)

$\square$

Lemma 5 **($T^+(G^o, \pi, \gamma, P)$)** *Let P, G, o, π and γ be as defined in Lemma 4. Suppose $CheckPoints(\gamma) = [p_1 \ldots p_n]$. Then $T^+(G^o, \pi, \gamma, P)$ satisfies the following:*

1. ***If** $P = p_i$ for some i then $T^+(G^o, \pi, \gamma, P)$ is the solution of the following linear programming problem : **maximize** $Var(\gamma, P)$ **subject to** $\mathcal{C}(G^o, \Gamma, \sqsubseteq)$*

2. ***If** the previous case does not apply and P is on line segment $[p_i, p_{i+1}]$ for some i then $T^+(G^o, \pi, \gamma, P)$ is the **minimum** of the following two items:*

   - $T^+(G^o, \pi, \gamma, p_i) + dist(P, p_i)/v^-(\gamma, p_i, p_{i+1})$

   - $T^+(G^o, \pi, \gamma, p_{i+1}) - dist(P, p_{i+1})/v^+(\gamma, p_i, p_{i+1})$

Proof The proof is very similar to the proof of Lemma 4 so it will be omitted. $\square$

Lemma 6 *Let G be a go theory, o be an object,π be a plan for G^o. Let γ be a movement in π such that $TCI(G^o, \pi, \gamma)$ is defined and t be any time point in $TCI(G^o, \pi, \gamma)$. Then*

(1) $P^-(G^o, \pi, \gamma, t) = P$ iff $T^+(G^o, \pi, \gamma, P) = t$.

(2) $P^+(G^o, \pi, \gamma, t) = P$ iff $T^-(G^o, \pi, \gamma, P) = t$.

Proof We are going to prove (1) as the proof for (2) is very similar to proof of (1).

If $P^-(G^o, \pi, \gamma, t) = P$ then in all instances of π at time t, o advanced to at least to P. Then there is no instance of π such that o is at P after t. Then by definition of $T^+(G^o, \pi, \gamma, P)$ we get $T^+(G^o, \pi, \gamma, P) \leq t$. By definition of $P^-(G^o, \pi, \gamma, t) = P$ we know that there is at least one instance $\mathcal{I}$ of π such that $\mathcal{I}(o,t) = P$ and $\mathcal{I}$ satisfies γ w.r.t. π in an interval including t. Hence $T^+(G^o,\pi,\gamma,P)$ can not be less than t. Thus $T^+(G^o,\pi,\gamma,P) = t$.

If $T^+(G^o,\pi,\gamma,P) = t$ then in all instances of π; o arrives P no later than t. Then there is no instance $\mathcal{I}$ of π such that o is at P after t hence for any time point after t the distance between $\mathcal{I}(o,t)$ and $loc_1(\gamma)$ is at least as much as the distance between P and $loc_1(\gamma)$. Then by definition of $P^-(G^o,\pi,\gamma,t)$ we get $P^-(G^o,\pi,\gamma,P) \in [P, loc_2(\gamma)]$. By definition of $T^+(G^o,\pi,\gamma,t) = P$ we know that there is at least one instance $\mathcal{I}$ of π such that $\mathcal{I}(o,t) = P$ and $\mathcal{I}$ satisfies γ w.r.t. π in an interval including t. Hence $P^-(G^o,\pi,\gamma,P)$ can not be further than P. Thus $P^-(G^o,\pi,\gamma,t) = P$. $\Box$

Corollary A-2 *Let G be a go theory, o be an object,π be a plan for G^o. Let γ be a movement in π such that $TCI(G^o,\pi,\gamma)$ is defined. Then*

(i) If $\mathcal{I}$ satisfies is γ w.r.t π over $[t_1,t_2]$ then $TCI(G^o,\pi,\gamma) \subset [t_1,t_2]$

(ii) If $\mathcal{I}$ is an instance of π then $\forall t \in TCI(G^o,\pi,\gamma)\ \mathcal{I}(o,t) \in LS(\gamma)$.

Proof Let $P = loc_1(\gamma)$ and $Q = loc_2(\gamma)$.

(i) By definition $TCI(G^o,\pi,\gamma) = [T^+(G^o,\pi,\gamma,P), T^-(G^o,\pi,\gamma,Q)]$. If

- $T^+(G^o,\pi,\gamma,P) < t_1$ then $T^+(G^o,\pi,\gamma,P)$ is not the latest time to be at P which contradicts the definition of $T^+(G^o,\pi,\gamma,P)$.
- $T^-(G^o,\pi,\gamma,Q) > t_2$ then $T^-(G^o,\pi,\gamma,Q)$ is not the earliest time to be at Q which contradicts the definition of $T^-(G^o,\pi,\gamma,Q)$.

(ii) It follows trivially from (i) and Definition 5 (satisfaction of a go atom over an interval)

□

Lemma 7 *Let G be a go theory, o be an object, π be a plan for G^o. Let γ be a movement in π such that $TCI(G^o, \pi, \gamma)$ is defined and t be any time point in $TCI(G^o, \pi, \gamma)$. If $\mathcal{I}$ is an instance of π, then $\mathcal{I}(o, t)$ is on the line segment $PCR(G^o, \pi, \gamma, t)$.*

Proof Let $\mathcal{I}(o, t) = Q$, it follows from corollary A-2 that $Q \in LS(\gamma)$.

Let $P^- = P^-(G^o, \pi, \gamma, t)$ and $P^+ = P^+(G^o, \pi, \gamma, t)$.

Then by definition of $P^-(G^o, \pi, \gamma, t)$, $\mathsf{dist}(loc_1(\gamma), P^-) \leq \mathsf{dist}(loc_1(\gamma), Q)$.

Similarly by definition of $P^+(G^o, \pi, \gamma, t)$, $\mathsf{dist}(loc_1(\gamma), P^+) \geq \mathsf{dist}(loc_1(\gamma), Q)$.

Since P^-, P^+ and Q are all on $LS(\gamma)$ we get $Q \in [P^-, P^+]$. Thus $\mathcal{I}(o, t)$ is on the line segment $PCR(G^o, \pi, \gamma, t)$. □

Lemma 8 *Let π be a plan for a go-theory G. Let γ be a movement of π such that $CheckPoints(\gamma) = [p_1, \ldots, p_n]$. Then for any i:$1 \leq i < n$, $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$ is equal to the following:*

1. $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$=$v^-(\gamma, p_i, p_{i+1})$ *when*

   (i) $T^-(G^o, \pi, \gamma, p_i) = T^+(G^o, \pi, \gamma, p_i)$ *and*

   (ii) $T^-(G^o, \pi, \gamma, p_{i+1}) = T^+(G^o, \pi, \gamma, p_{i+1})$ *and*

   (iii) $v^-(\gamma, p_i, p_{i+1}) = \frac{\mathsf{dist}(p_i, p_{i+1})}{T^+(G^o, \pi, \gamma, p_{i+1}) - T^+(G^o, \pi, \gamma, p_i)}$

2. $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$=$v^+(\gamma, p_i, p_{i+1})$ *otherwise*

Suppose $[P, Q]$ *is a line segment such that* $[P, Q] \subseteq LS(\gamma)$. *Then,*

$$V^+(G^o, \pi, \gamma, P, Q) = max\{V^+(G^o, \pi, \gamma, p_i, p_{i+1}) \mid [P, Q] \cap [p_i, p_{i+1}] \textit{ is line segment } \}$$

Proof The last part of the lemma is trivial for any subsegment $[P, Q]$ of $LS(\gamma)$, by definition of maximum speed (Defintion 26), $V^+(G^o, \pi, \gamma, P, Q)$ has to be the maximum of the maximum speed on any subsegment of $[P, Q]$. As shown in Lemma 2 between every consecutive check points the object is subject to different speed constraints that are independent form the other check points. Thus $V^+(G^o, \pi, \gamma, P, Q)$ has to be equal to the maximum of $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$ for any $[p_i, p_{i+1}]$ that shares a common segment with $[P, Q]$.

Now we are going to show that for any consecutive check points p_i and p_{i+1}, the maximum speed on $[p_i, p_{i+1}]$, $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$, is equal to (1) or (2) depending on the conditions (i), (ii) and (iii). For readability let's assume $T^-(G^o, \pi, \gamma, p_i) = T_i^-$, $T^-(G^o, \pi, \gamma, p_{i+1}) = T_{i+1}^-$, $T^+(G^o, \pi, \gamma, p_i) = T_i^+$ and $T^+(G^o, \pi, \gamma, p_{i+1}) = T_{i+1}^+$,

Assume the conditions (i), (ii) and (iii) are all true then

(iv) By (i), in all models the object arrives at p_i at T_i^-

(v) By (ii), in all models the object arrives at p_{i+1} at T_{i+1}^-

(vi) By (iii),(iv) and (v), in all models the average speed of the object, between T_i^- and T_{i+1}^- is equal to $v^-(\gamma, p_i, p_{i+1})$.

(vii) It follows from Lemma 2 that in any model the speed of the object can not be less than $v^-(\gamma, p_i, p_{i+1})$ at any time point in $[T_i^-, T_{i+1}^-]$. Thus if at any time point the speed of the object is more than $v^-(\gamma, p_i, p_{i+1})$ then the average speed between T_i^- and T_{i+1}^- has to be more than $v^-(\gamma, p_i, p_{i+1})$ which contradicts (vi). Hence the maximum speed on the segment $[p_i, p_{i+1}]$ has to be $v^-(\gamma, p_i, p_{i+1})$.

This proves that if conditions (i), (ii) and (iii) are all true then $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$, is equal to (1). Next we are going to show that if any of (i), (ii) and (iii) is not true than there is an instance of π such that $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$, is equal to (2). There are three cases to consider:

(a) When (i) is false, $T_i^- < T_i^+$.

As proven in Lemma 4 there is an instance of π, $\mathcal{I}$, such that $\mathcal{I}(o, T_i^-) = p_i$ and $\mathcal{I}(o, T_{i+1}^-) = p_{i+1}$. Similarly there is an instance of π, $\mathcal{I}'$, such that $\mathcal{I}'(o, T_i^+) = p_i$ and $\mathcal{I}'(o, T_{i+1}^+) = p_{i+1}$. Assume for $\mathcal{I}$ the average speed of o during $[T_i^-, T_{i+1}^-]$ is more than $v^-(\gamma, p_i, p_{i+1})$. Then it is possible for $\mathcal{I}$ to have a constant speed of $v^+(\gamma, p_i, p_{i+1})$ for some time and then switch to constant speed of $v^-(\gamma, p_i, p_{i+1})$ for the rest of the time in a way to achieve the desired average speed. Hence there can be a model in which o's speed is as fast as $v^+(\gamma, p_i, p_{i+1})$. We can reach the same conclusion for $\mathcal{I}'$ if the average speed of o in $\mathcal{I}'$ during $[T_i^-, T_{i+1}^-]$ is more than $v^-(\gamma, p_i, p_{i+1})$.

We still need to prove the case when for both $\mathcal{I}$ and $\mathcal{I}'$ the average speed during $[T_i^-, T_{i+1}^-]$ is $v^-(\gamma, p_i, p_{i+1})$. Let $v = \frac{\mathsf{dist}(p_i, p_{i+1})}{T_{i+1}^- - T_i^+)}$. It is clear that $v < v^-(\gamma, p_i, p_{i+1})$. Once again there are two cases to consider:

- Case 1: $v^+(\gamma, p_i, p_{i+1}) \leq v$. Then we can use the same method above to construct an instance of π, $\mathcal{I}''$ such that $\mathcal{I}''(o, T_i^+) = p_i$ and $\mathcal{I}''(o, T_{i+1}^-) = p_{i+1}$. Hence (2) is satisfied.

- Case 2: $v^+(\gamma, p_i, p_{i+1}) > v$, then there is a t such that $T_{i+1}^- < t < T_{i+1}^+$ and

$v^+(\gamma, p_i, p_{i+1}) \leq \frac{\mathsf{dist}(p_i,p_{i+1})}{T^-_{i+1}-t}$. Once again we can use the same method above to construct an instance of π, $\mathcal{I}^*$ such that $\mathcal{I}^*(o,t) = p_i$ and $\mathcal{I}^*(o,T^-_{i+1}) = p_{i+1}$. Hence (2) is satisfied.

(b) When (ii) is false, $T^-_{i+1} < T^+_{i+1}$. We can show that $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$, is equal to (2) using the same arguments in [a].

(c) When (iii) is false We can show that $V^+(G^o, \pi, \gamma, p_i, p_{i+1})$, is equal to (2) using the first paragraph in proof of (a).

□

Lemma 9 *Let π be a plan for a go-theory G. Let γ be a movement of π such that $CheckPoints(\gamma) = [p_1, \ldots, p_n]$. Then for any i:$1 \leq i < n$, $V^-(G^o, \pi, \gamma, p_i, p_{i+1})$ is equal to the following:*

1. $V^-(G^o, \pi, \gamma, p_i, p_{i+1})$=$v^+(\gamma, p_i, p_{i+1})$ *when*

   (i) $T^-(G^o, \pi, \gamma, p_i) = T^+(G^o, \pi, \gamma, p_i)$ *and*

   (ii) $T^-(G^o, \pi, \gamma, p_{i+1}) = T^+(G^o, \pi, \gamma, p_{i+1})$ *and*

   (iii) $v^+(\gamma, p_i, p_{i+1}) = \frac{\mathsf{dist}(p_i,p_{i+1})}{T^+(G^o,\pi,\gamma,p_{i+1})-T^+(G^o,\pi,\gamma,p_i)}$

2. $V^-(G^o, \pi, \gamma, p_i, p_{i+1})$=$v^-(\gamma, p_i, p_{i+1})$ *otherwise*

Suppose $[P, Q]$ is a line segment such that $[P, Q] \subseteq LS(\gamma)$. Then,

$$V^-(G^o, \pi, \gamma, P, Q) = min\{V^-(G^o, \pi, \gamma, p_i, p_{i+1}) \mid [P, Q] \cap [p_i, p_{i+1}] \textit{ is line segment } \}$$

Proof The proof is very similar to the proof of Lemma 8 so it will be omitted. □

Answering Ground Atomic Queries

Ground Atomic go Queries

Lemma 10 *Let o be an object, $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a go-atom and G be a go-theory. Suppose π is a plan for G^o. All instances of π satisfy g iff all the following conditions hold:*

1. *There is a movement γ in π that is relevant to g*
2. $t_1^- \leq T^-(G^o, \pi, \gamma, P_1) \leq T^+(G^o, \pi, \gamma, P_1) \leq t_1^+$
3. $t_2^- \leq T^-(G^o, \pi, \gamma, P_2) \leq T^+(G^o, \pi, \gamma, P_2) \leq t_2^+$
4. $v^- \leq V^-(\gamma, P_1, P_2) \leq V^+(\gamma, P_1, P_2) \leq v^+$.

Proof The lemma has two parts.

If part: If all the instances of π satisfy g then the conditions hold.

Since in all instances of π the only time the object is forced to move on a line segment with a specific direction is during the satisfaction of a movement, the existence of at least one movement related to g is obvious.

If there is only one related movement than conditions 2,3 and 4 follow from definitions: satisfaction of a go atom, earliest/latest arrival time and min/max speed. For example assume condition (2) does not hold. Then there is an instance of π, say $\mathcal{I}$ that travels on $[P_1, P_2]$ during $[t_1, t_2]$ such that t_1 is before t_1^- or after t_1^+ so $\mathcal{I}$ does not satisfy g over $[t_1, t_2]$. Since γ is the only movement related to g then in $\mathcal{I}$ can avoid satisfying g in another time interval as well. Thus it will not satisfy g, contradicting the initial assumption. Similar reasoning can be used to prove conditions (3) and (4) must hold as well.

If there is more than one related movement then we need to show at least for one of them the conditions 2 to 4 must hold. We are going to illustrate this using two related movements γ and γ' such that γ is ordered before γ' in π. Assume for both movements either condition 2 or 3 does not hold. We have the following cases:

a) Condition 2 does not hold for both movements. Then the subcases are:

   - $t_1^- > T^-(G^o, \pi, \gamma, P_1)$ and $t_1^- > T^-(G^o, \pi, \gamma', P_1)$. Then there is an instance $\mathcal{I}$ of π such that while satisfying γ', o travels on $[P_1, P_2]$ during $[t_1, t_2]$ where $t_1 = T^-(G^o, \pi, \gamma', P_1)$ and t_1 is before t_1^-. Hence $\mathcal{I}$ does not satisfy g over $[t_1, t_2]$. $\mathcal{I}$ can avoid satisfying g in another time interval after t_2 because there is no other movement after γ' that is related to g. Furthermore it can not satisfy g before t_1 since t_1 is already before t_1^-.
   - $t_1^- > T^-(G^o, \pi, \gamma, P_1)$ and $t_1^+ > T^+(G^o, \pi, \gamma', P_1)$. Then there is an instance $\mathcal{I}_1$ of π such that while satisfying γ, o travels on $[P_1, P_2]$ during $T_1 = [T^-(G^o, \pi, \gamma, P_1), t_2]$. Hence $\mathcal{I}_1$ does not satisfy g over T_1. There is also an instance $\mathcal{I}_2$ of π such that while satisfying γ', o travels on $[P_1, P_2]$ during $T_2 = [T^+(G^o, \pi, \gamma', P_1), t_2']$. Hence $\mathcal{I}_2$ does not satisfy g over T_2. Then there is another instance of π, say $\mathcal{I}$ that while satisfying γ, o travels on $[P_1, P_2]$ during T_1 and while satisfying γ', o travels on $[P_1, P_2]$ during T_2. This is possible because first one requires earliest start of γ and second one requires latest start of γ' and these two requirements cannot contradict each other. Clearly $\mathcal{I}$ does not satisfy g during γ or γ' thus it can avoid satisfying g in other times.

- $t_1^+ > T^+(G^o, \pi, \gamma, P_1)$. Then there is an instance $\mathcal{I}$ of π such that while satisfying γ, o travels on $[P_1, P_2]$ during $[t_1, t_2]$ where $t_1 = T^+(G^o, \pi, \gamma, P_1)$ and t_1 is after t_1^+. Hence $\mathcal{I}$ does not satisfy g over $[t_1, t_2]$. $\mathcal{I}$ can avoid satisfying g in another time interval after t_2 because there is no other movement before γ that is related to g. Furthermore it can not satisfy g after t_2 since t_1 is already after t_1^+.

b) Condition 2 does not hold for γ and condition 3 does not hold for γ'

- $t_1^- > T^-(G^o, \pi, \gamma, P_1)$ and $t_2^- > T^-(G^o, \pi, \gamma', P_2)$. Then there is an instance $\mathcal{I}_1$ of π such that while satisfying γ, o travels on $[P_1, P_2]$ during $T_1 = [T^-(G^o, \pi, \gamma, P_1), t_2]$. Hence $\mathcal{I}_1$ does not satisfy g over T_1. There is also an instance $\mathcal{I}_2$ of π such that while satisfying γ', o travels on $[P_1, P_2]$ during $T_2 = [t1, T^-(G^o, \pi, \gamma', P_2)]$. Hence $\mathcal{I}_2$ does not satisfy g over T_2. Then there is another instance of π, say $\mathcal{I}$ that while satisfying γ, o travels on $[P_1, P_2]$ during T_1 and while satisfying γ', o travels on $[P_1, P_2]$ during T_2. This is possible because first one requires earliest start of γ (consequently allows earliest end time of γ) and second one requires earliest termination of γ' (possible when γ ends earliest allowing the earliest start of γ') and these two requirements can be combined without contradicting each other. Clearly $\mathcal{I}$ does not satisfy g during γ or γ' thus it can avoid satisfying g in other times.
- $t_1^- > T^-(G^o, \pi, \gamma, P_1)$ and $t_2^+ > T^+(G^o, \pi, \gamma', P_2)$. Using similar reasoning we can show the existence of an instance of π, say $\mathcal{I}$ that while satisfying γ, o travels on $[P_1, P_2]$ during $T_1 = [T^-(G^o, \pi, \gamma, P_1), t_2]$ and while satisfying

γ', o travels on $[P_1, P_2]$ during $T_2 = [t1, T^+(G^o, \pi, \gamma', P_2)]$. This is possible because first one requires earliest start of γ and second one requires latest termination of γ' and these two requirements can be combined without contradicting each other. Clearly $\mathcal{I}$ does not satisfy g during γ or γ' thus it can avoid satisfying g in other times.

– $t_1^+ > T^+(G^o, \pi, \gamma, P_1)$. Same as explained in case (a3)

c) Condition 3 does not hold for γ and condition 2 does not hold for γ'

– $t_2^- > T^-(G^o, \pi, \gamma, P_2)$ and $t_1^- > T^-(G^o, \pi, \gamma', P_1)$. Same as in case (a1).

– $t_2^- > T^-(G^o, \pi, \gamma, P_2)$ and $t_1^+ > T^+(G^o, \pi, \gamma', P_1)$. Using similar reasoning we can show the existence of an instance of π, say $\mathcal{I}$ that while satisfying γ, o travels on $[P_1, P_2]$ during $T_1 = [t_1, T^-(G^o, \pi, \gamma, P_2)]$ and while satisfying γ', o travels on $[P_1, P_2]$ during $T_2 = [T^+(G^o, \pi, \gamma', P_1), t_2]$. This is possible because first one requires termination start of γ and second one requires latest start of γ' and these two requirements can be combined without contradicting each other. Clearly $\mathcal{I}$ does not satisfy g during γ or γ' thus it can avoid satisfying g in other times.

– $t_2^+ > T^+(G^o, \pi, \gamma, P_2)$. Same as explained in case (a3)

d) Condition 3 does not hold for both movements This case is symmetric to case (a).

Only if part: If all the conditions hold then all the instances of π satisfy g. This part is easy.

i By condition (1) we know that γ is related to g thus in all instances of π there is a time interval such that γ is satisfied hence there a time interval $[t_1, t_2]$ such that o moves on the line segment $[P_1, P_2]$ from P_1 to P_2.

ii By definition of earliest/latest arrival times and (i) we have $T^-(G^o, \pi, \gamma, P_1) \leq t_1 \leq T^+(G^o, \pi, \gamma, P_1)$.

iii By condition (2) and (ii) we have $t_1^- \leq t_1 \leq t_1^+$.

iv Similarly by condition (3) and definition of earliest/latest arrival times we have $t_2^- \leq t_2 \leq t_2^+$.

v By definition of max/min speed and (i) for any instance $\mathcal{I}$ of π we have $V^-(\gamma, P_1, P_2) \leq d(\mathcal{I}(o,t))/d(t) \leq V^+(\gamma, P_1, P_2)$ when $t \in [t_1, t_2]$.

vi By condition 4 and (v) for any instance $\mathcal{I}$ of π we have $v^- \leq d(\mathcal{I}(o,t))/d(t) \leq v^+$ when $t \in [t_1, t_2]$.

vii By (i),(iii), (iv) and (vi) and definition of satisfaction of a go atom, every instance of π satisfies g.

□

Theorem 7 *Suppose G is a consistent* go*-theory and $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ is a ground* go *atom. Then: g is a logical consequence of G iff for every plan π for G^o* CheckGo$(G, \Gamma, \sqsubseteq, g)$ *returns "true".*

Proof It is easy to see that Algorithm CheckGo(G, π, g) returns true only when the conditions in Lemma 10 are all true.

If part: If $G \models g$ then for every plan π for G^o CheckGo(G, π, g) returns "true". Assume there is a plan π for G^o such that algorithm returns false. Then by Lemma 10 there is an interpretation $\mathcal{I}$ that is an instance of π and $I \not\models g$. By definition of instance (Definition 21), $I \models G$. But this contradicts $G \models g$.

Only if part: If for every plan π for G^o $\textsf{CheckGo}(G, \pi, g)$ returns "true" then $G \models g$. Assume $G \not\models g$. Then there is a model of G, $\mathcal{I}^*$ such that $\mathcal{I}^* \not\models g$. As shown in Theorem 2, for every model $\mathcal{I}'$ of G there is a plan π' such that $\mathcal{I}'$ is an instance of π'. Assume $\mathcal{I}^*$ is an instance of π^*. Then by Lemma 10, $\textsf{CheckGo}(G, \pi^*, g)$ returns false but this contradicts the initial assumption of for every plan π for G^o $\textsf{CheckGo}(G, \pi, g)$ returns "true". Hence $\mathcal{I}^*$ can not exist. □

Ground Atomic in Queries

Lemma 11 *Suppose $g = \textsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^+, t_2^+, v^-, v^+)$ is a* go*-atom, $G = \{g\}$ is a* go*-theory and $a = \textsf{in}(o, Q_1, Q_2, t_1, t_2)$ is a ground* in()*-atom. a is a logical consequence of G iff the following conditions hold:*

a) $Vol(a)$ intersects the line segment $LS(g)$

b) o's latest arrival time at point P_1', $T^+(g, P_1') \leq t_2$

c) o's earliest arrival time at point P_2', $T^-(g, P_2') \geq t_1$

Where P_1' and P_2' are points on $LS(g)$ such that $L = [P_1', P_2']$ is the longest sub-segment of $LS(g)$ inside $Vol(a)$ and $\textsf{dist}(loc_1(g), P_1') \leq \textsf{dist}(loc_1(g), P_2')$.

Proof If a is a logical consequence of G then there conditions (a),(b) and (c) hold.

- Assume $Vol(a)$ does not intersect $LS(g)$. Then one can create an interpretation $\mathcal{I}$ such that $\mathcal{I} \models g$ and $\forall t \in [t_1, t_2]$ $\mathcal{I}(o, t) = P \wedge P \notin Vol(q)$. This is possible because only constraint on $\mathcal{I}$ is to place the object on $LS(g)$ (which is outside of $Vol(a)$) in the time interval $[t_1^-(g), t_2^+(g)]$. Thus if $Vol(a)$ does not intersect $LS(g)$ then $\exists \mathcal{I} \models g$ such that $\mathcal{I} \not\models q$. This is a contradiction.

- If there is a model $\mathcal{I}$ of g such that arrives P_1' later than t_2, then it will be inside $Vol(a)$ after t_2 in which case a will be false. This will contradict the assumption that a is a logical consequence of g. Hence $T^+(g, P_1') \leq t_2$
- Similarly if a model of g lets the object arrive P_2' before t_1 than the object passed through $Vol(a)$ before t_1 and the same model can place the object outside of $Vol(a)$ until t_2. This will contradict the assumption that a is a logical consequence of g. Thus $T^-(g, P_2') \geq t_1$

If the conditions (a),(b) and (c) hold then a is a logical consequence of G. Assume a is not a logical consequence of G, then this implies

$$\text{(i) } \exists \mathcal{I} \models g \text{ such that } \forall t \in [t_1, t_2]\, \mathcal{I}(o, t) = P \notin Vol(a).$$

Then by definition of $\mathcal{I} \models g$ and condition (a) we get,

$$\text{(ii) } \exists T_1, T_2 \text{ such that } T_1 < T_2 \wedge \mathcal{I}(o, T_1) = P_1' \wedge \mathcal{I}(o, T_2) = P_2'.$$

$$\text{(iii) } \forall\, t,\ T_1 \leq t \leq T_2, \mathcal{I}(o, t) \in Vol(a).$$

Combining (i) and (iii) we get

$$\text{(iv) } T_1 < T_2 < t_1 \vee t_2 < T_1 < T_2.$$

By definition of earliest/latest arrivals and (iv) we get

$$\text{(v) } T^-(g, P_2') \leq T_2 < t_1 \vee t_2 < T_1 \leq T^+(g, P_1').$$

(v) contradicts (b) and (c) hence interpretation $\mathcal{I}$ can not exist. $\square$

Corollary A-3 *Suppose* $a = \mathsf{in}(o, Q_1, Q_2, t_1, t_2)$ *is a ground* $\mathsf{in}()$*-atom and* G *is a* go-*theory,* π *is a plan for* G^o*. If* $G \models a$ *then there is a movement* γ *in* π *that is related to* a*.*

Proof Let $\gamma_1 \sqsubseteq \gamma_2 \ldots \gamma_n$ be the movements of π and $\mathcal{I}$ be an instance of π. Then by definition of an instance there exists time intervals $[t_{11}t_{12}], [t_{21}t_{22}] \ldots [t_{n1}t_{n2}]$ such that $\mathcal{I}$ satisfies γ_i over $[t_{i1}t_{i2}]$ and $t_{i2} \leq t_{j1}$ whenever $\gamma_i \sqsubseteq \gamma_j$. Furthermore let for all t such that $t \in [t_1, t_2]$ and $\not\exists\, i \mid \;\; t \in [t_{i1}t_{i2}]$, $\mathcal{I}(o, t) = P \mid P \notin Vol(a)$. This assumption is valid since $\mathcal{I}(o, t)$ can be any point when not satisfying a go atom. Assume there is no movement γ of π such that γ is related to a then it follows from movement satisfaction definition that

$$\text{(i)}\ \forall\, \gamma_i \mid \quad [t_1, t_2] \cap [t_{i1}t_{i2}] \neq \emptyset \implies extent(\gamma_i) \cap [t_1, t_2] \neq \emptyset$$

Using (i) and definition of a related movement and we get:

$$\text{(ii)}\ \forall\, \gamma_i \mid \quad [t_1, t_2] \cap [t_{i1}t_{i2}] \neq \emptyset \implies LS(\gamma_i) \cap Vol(a) = \emptyset$$

Combining (ii) and $\mathcal{I}$ we get

$$\text{(iii)}\ \forall\, i, t \in [t_1, t_2] \cap [t_{i1}t_{i2}], I(o, t) \notin Vol(a)$$

Finally from (iii) and $\mathcal{I}$ it folows:

$$\text{(iv)}\ \forall t \in [t_1, t_2], I(o, t) \notin Vol(a)$$

But (iv) entails that $\mathcal{I} \not\models a$ which contradicts that $G \models a$ and $\mathcal{I} \models G$. Thus there has to be a movement in π that is related to a. $\square$

Lemma 12 *Suppose $a = \mathsf{in}(o, Q_1, Q_2, t_1, t_2)$ is a ground $\mathsf{in}()$-atom and G is a* go-*theory, π is a plan for G^o. All instances of π satisfy a iff there is a movement γ in π such that all the following conditions hold:*

(i) γ *is related to* a

(ii) $T^+(G^o, \pi, \gamma, P_1) \leq t_2$

(iii) $T^-(G^o, \pi, \gamma, P_2) \geq t_1$

Where P_1 and P_2 are points on $LS(\gamma)$ such that $L = [P_1, P_2]$ is the longest sub-segment of $LS(\gamma)$ inside $Vol(a)$ and $\mathsf{dist}(loc_1(\gamma), P_1) \leq \mathsf{dist}(loc_1(\gamma), P_2)$.

Proof The lemma has two parts.

If part: If all the instances of π satisfy a then the conditions hold.

It follows from Corollary A-3 that (i) holds. So there is at least one movement in π that is related to a. Assume for all related movements, (ii) or (iii) do not hold. Let S_1 and S_2 be two sets of movements such that:

$S_1 = \{\gamma \mid$ γ is a movement in π and is related to a and does not satisfy (ii) $\}$ and

$S_2 = \{\gamma \mid$ γ is a movement in π and is related to a and does not satisfy (iii) $\}$.

Let $\gamma_1 \sqsubseteq \gamma_2 \ldots \gamma_n$ be the ordered elements of $S_1 \cup S_2$ which are the all the movements in π that are related to a. We are going to examine three cases:

- If $\gamma_1 \in S_1$ then by definition of maximum arrival time there is an instance of π such that $\mathcal{I}(o, t) = P_1$ and $\mathcal{I} \models^T \gamma_1$ and $t \in T$ and $t > t_2$. Since none of the movements in π that are before γ_1 are not related to a and the portion of $LS(\gamma_1)$

that is inside $Vol(a)$ is going to be visited after t_2, in the instance $\mathcal{I}$, it is possible for the object to stay out of $Vol(a)$ throughout the time interval $[t_1, t_2]$. Then $\mathcal{I} \not\models a$ which contradicts the initial assumption that all instances of π satisfy a.

- If $\gamma_n \in S_2$ then by definition of minimum arrival time there is an instance of π such that $\mathcal{I}(o,t) = P_2$ and $\mathcal{I} \models^T \gamma_n$ and $t \in T$ and $t < t_1$. Since none of the movements in π that are after γ_n are not related to a and the portion of $LS(\gamma_n)$ that is inside $Vol(a)$ has to be visited before t_1, in the instance $\mathcal{I}$, it is possible for the object to stay out of $Vol(a)$ throughout the time interval $[t_1, t_2]$. Then $\mathcal{I} \not\models a$ which contradicts the initial assumption that all instances of π satisfy a.

- Let γ_i be the first movement in S_1 such that $i > 1$. Then $\gamma_{i-1} \in S_2$. By definition of maximum arrival time there is an instance of π such that $\mathcal{I}_1(o,t) = P_1^i$ and $\mathcal{I}_1 \models^T \gamma_i$ and $t \in T$ and $t > t_2$. Similarly by definition of minimum arrival time there is an instance of π such that $\mathcal{I}_2(o,t') = P_2^{i-1}$ and $\mathcal{I}_2 \models^{T'} \gamma_{i-1}$ and $t' \in T'$ and $t' < t_1$. Note that $t' < t_1 \leq t_2 < t$. It is easy to see that using $\mathcal{I}_1$ and $\mathcal{I}_2$ we can construct another instance of π, $\mathcal{I}_3$ such that in $\mathcal{I}_3$, the object stays out of $Vol(a)$ throughout the time interval $[t_1, t_2]$. Then $\mathcal{I}_3 \not\models a$ which contradicts the initial assumption that all instances of π satisfy a.

This proves the if-part of the lemma.

Only-if part: If all the conditions hold then all the instances of π satisfy a.

1. By condition (i) the a path of the movement γ and the query volume has a nonempty intersection, which is a line segment from P_1 to P_2, i.e. $[P1, P2] = LS(\gamma) \cap Vol(a)$
2. By definition of an instance: for every instance $\mathcal{I}$ of π there is a time interval T over

which $\mathcal{I}$ satisfies γ. Then there are time points T_1 and T_2 such that $\mathcal{I}(o, T_1) = P_1$ and $\mathcal{I}(o, T_1) = P_2$ and $[T_1, T_2] \in T$. It follows from satisfaction of a movement that $\forall t \in [T_1, T_2], I(o, t) \in [P_1, P_2]$.

3. By definition of earliest/latest arrival times and (2);

   $T_1 \leq T^+(G^o, \pi, \gamma, P_1)$ and $T_2 \geq T^-(G^o, \pi, \gamma, P_2)$

4. By (3) and conditions (ii) and (iii) we have:

   $T_1 \leq T^+(G^o, \pi, \gamma, P_1) \leq t_2$ and $T_2 \geq T^-(G^o, \pi, \gamma, P_2) \geq t_1$.

   It is trivial to see that neither $[t_1, t_2]$ nor $[T_1, T_2]$ can terminate before the other one starts. Thus the intervals $[t_1, t_2], [T_1, T_2]$ have a non empty intersection for any instance of π. Furthermore by (2), during $[T_1, T_2]$ the object is always in the query volume. Hence all instances of π satisfy a.

□

Theorem 8 *Suppose G is a consistent* go-*theory and $a = \mathsf{in}(o, P_1, P_2, t_1, t_2)$ is a ground atom. Then: a is a logical consequence of G iff for every plan π for G^o algorithm* $\mathsf{CheckIn}(G, o, \pi, a)$ *returns "true".*

Proof It is easy to see that Algorithm $\mathsf{CheckIn}(G, \pi, a)$ returns true only when the conditions in Lemma 12 are all true.

If part: If $G \models a$ then for every plan π for G^o $\mathsf{CheckIn}(G, \pi, a)$ returns "true". Assume there is a plan π for G^o such that algorithm returns false. Then by Lemma 12 there is an interpretation $\mathcal{I}$ that is an instance of π and $I \not\models a$. By definition of instance (Definition 21), $I \models G$. But this contradicts $G \models a$.

Only if part: If for every plan π for G^o $\mathsf{CheckIn}(G, \pi, a)$ returns "true" then $G \models a$. Assume $G \not\models a$. Then there is a model of G, $\mathcal{I}^*$ such that $\mathcal{I}^* \not\models a$. As shown in Theorem 2, for every model $\mathcal{I}'$ of G there is a plan π' such that $\mathcal{I}'$ is an instance of π'. Assume $\mathcal{I}^*$ is an instance of π^*. Then by Lemma 12, $\mathsf{CheckIn}(G, \pi^*, a)$ returns false but this contradicts the initial assumption of for every plan π for G^o $\mathsf{CheckIn}(G, \pi, a)$ returns "true". Hence $\mathcal{I}^*$ can not exist. $\square$

Ground atomic near queries

Lemma 13 *Let $G = \{g_1, g_2\}$ be a go theory such that $obj(g_1) = o_1$ and $obj(g_2) = o_2$. $G \models \mathsf{near}(o_1, o_2, d, t, t)$ iff:*

1. *$TCI(g_1)$ and $TCI(g_2)$ are defined and*
2. *$t \in TCI(g_1)$ and $t \in TCI(g_2)$ and*
3. *$\mathsf{dist}(P^-(g_1, t), P^-(g_2, t)) \leq d$ and*
4. *$\mathsf{dist}(P^-(g_1, t), P^+(g_2, t)) \leq d$ and*
5. *$\mathsf{dist}(P^+(g_1, t), P^-(g_2, t)) \leq d$ and*
6. *$\mathsf{dist}(P^+(g_1, t), P^+(g_2, t)) \leq d$.*

Proof **Only if part:** Show that when these conditions hold $G \models \mathsf{near}(o_1, o_2, d, t, t)$. If $t \in TCI(g_1)$ then by Lemma 7, for every model $\mathcal{I}$ of G, $\mathcal{I}(o_1, t)$ is on the line segment $PCR(g_1, t)$. Similarly if $t \in TCI(g_2)$ then for every model $\mathcal{I}$ of G, $\mathcal{I}(o_2, t)$ is on the line segment $PCR(g_2, t)$. By definition of positional certainty interval; $PCR(g_1, t) = [P^-(g_1, t), P^+(g_1, t)]$ and $PCR(g_2, t) = [P^-(g_2, t), P^+(g_2, t)]$. The maximum distance between two line segments is achieved at one of the end points of each line segments.

Since conditions (3) to (6) are true; the maximum distance between $PCR(g_1, t)$ and $PCR(g_2, t)$ is less than d. Hence in any model $\mathcal{I}$ of G the distance between $\mathcal{I}(o_1, t)$ and $\mathcal{I}(o_2, t)$ is less than or equal to d. Thus every model of G also satisfies $\mathsf{near}(o_1, o_2, d, t, t)$

If part: Show that if $G \models \mathsf{near}(o_1, o_2, d, t, t)$ then all the conditions are satisfied.

(i) Assume $TCI(g_1)$ is not defined. Then by Definition 24, we have:

$T^+(g_1, loc_1(g_1)) \geq T^-(g_1, loc_2(g_1))$. By definition of earliest and latest arrival times there is an interpretation $\mathcal{I}_1$ that satisfy g_1 over $[T^+(g_1, loc_1(g_1)), t_2]$, for some t_2. Similarly there is an interpretation $\mathcal{I}_2$ that satisfy g_1 over $[t_1, T^-(g_1, loc_2(g_1))]$, for some t_1. If $t < T^+(g_1, loc_1(g_1))$ then $I_1(o_1, t)$ can be any point that is more than d distant to $\mathcal{I}_1(o_2, t)$. If $t > T^-(g_1, loc_2(g_1))$ then $I_2(o_1, t)$ can be any point that is more than d distant to $\mathcal{I}_2(o_2, t)$.Thus either $\mathcal{I}_1 \not\models \mathsf{near}(o_1, o_2, d, t, t)$ or $\mathcal{I}_2 \not\models \mathsf{near}(o_1, o_2, d, t, t)$, which contradicts the assumption that $G \models \mathsf{near}(o_1, o_2, d, t, t)$.

As a result of similar reasoning $TCI(g_2)$ should be defined and (1) has to hold.

(ii) Assuming (2) is not true leads to a contradiction using the similar arguments in (i).

(iii) Assume (3) is not true. By definition of minimal advancement point there is a model of G, say $\mathcal{I}$ such that $\mathcal{I}(o_1, t) = P^-(g_1, t)$ and $\mathcal{I}(o_2, t) = P^-(g_2, t)$. Then the distance between o_1 and o_2 is more than d hence $\mathcal{I} \not\models \mathsf{near}(o_1, o_2, d, t, t)$, which contradicts the assumption that $G \models \mathsf{near}(o_1, o_2, d, t, t)$.

We can use the same arguments to show that (4), (5) and (6) are also true.

□

Lemma 14 *Let $G = \{g_1, g_2\}$ be a go theory such that $obj(g_1) = o_1$ and $obj(g_2) = o_2$. $G \models \mathsf{near}(o_1, o_2, d, t_1, t_2)$ iff:*

1. $TCI(g_1)$ *and* $TCI(g_2)$ *are defined and*

2. $[t_1, t_2] \in TCI(g_1)$ *and* $[t_1, t_2] \in TCI(g_2)$ *and*

3. $\forall t \in [t_1, t_2]\ \mathsf{dist}(P^-(g_1, t), P^-(g_2, t)) \leq d$ *and*

4. $\forall t \in [t_1, t_2]\ \mathsf{dist}(P^-(g_1, t), P^+(g_2, t)) \leq d$ *and*

5. $\forall t \in [t_1, t_2]\ \mathsf{dist}(P^+(g_1, t), P^-(g_2, t)) \leq d$ *and*

6. $\forall t \in [t_1, t_2]\ \mathsf{dist}(P^+(g_1, t), P^+(g_2, t)) \leq d$.

Proof It follows from Lemma 13 and Definition 5 (satisfaction of a near atom) that $G \models \mathsf{near}(o_1, o_2, d, t_1, t_2)$ if and only if for every time point t in $[t_1, t_2]$ Lemma 13 holds. If we aggregate the conditions in Lemma 13, we get exactly the conditions in Lemma 14.

□

Lemma 15 *Let* $g = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ *be a normalized go atom such that* $TCI(g)$ *is defined. If* t *is a time point in* $TCI(g)$*, then*

$$P^-(g,t) = \begin{cases} P_1 + v^-(t - t_1^+)\vec{g} & \text{if } t < t^* \\ P_2 + v^+(t - t_2^+)\vec{g} & \text{if } t^* \leq t \end{cases}$$

$$P^+(g,t) = \begin{cases} P_1 + v^+(t - t_1^-)\vec{g} & \text{if } t < T^* \\ P_2 + v^-(t - t_2^-)\vec{g} & \text{if } T^* \leq t \end{cases}$$

where $t^* \in \mathbf{R}$ *such that* $P_1 + v^-(t^* - t_1^+)\vec{g} = P_2 + v^+(t^* - t_2^+)\vec{g}$ *and* $T^* \in \mathbf{R}$ *such that* $P_1 + v^+(T^* - t_1^-)\vec{g} = P_2 + v^-(T^* - t_2^-)\vec{g}$.

Proof The proof is very similar to proof of Lemma 4. Because g is normalized we know that there are two models of g, $\mathcal{I}_1$ and $\mathcal{I}_2$ that satisfy g over time intervals $[t_1^-, t]$ and $[t', t_2^-]$

respectively. As shown in proof of Lemma 4 there is an interpretation I that satisfies g over $[t_1^-, t_2^-]$. Using $\mathcal{I}_1$ and $\mathcal{I}_2$ we can define a $\mathcal{I}$ as follows:

$$\mathcal{I}(o,t) = \begin{cases} \mathcal{I}_1(o,t) & \text{if } t \leq t_1^- \\ P_1 + v^+(t - t_1^-)\vec{g} & \text{if } t_1^- < t < t_2^- \wedge t < t^* \\ P_2 - v^-(t_2^- - t)\vec{g} & \text{if } t_1^- < t < t_1^- \wedge t^* \leq t \\ \mathcal{I}_2(o,t) & \text{if } t \geq t_2^- \end{cases}$$

where $t^* \in \mathbf{R}$ such that $P_1 + v^+(t^* - t_1^-)\vec{g} = P_2 - v^-(t_2^- - t^*)\vec{\gamma}$.

As shown in Lemma 4, $\mathcal{I} \models g$ and for every $P \in [P_1, P_2]$, $\mathcal{I}(o, T^-(g,P)) = P$. It follows from Lemma 6, that $T^-(g,P)$ and $P^+(g,t)$ are inverse functions hence $P^+(g,t) = \mathcal{I}(o,t)$. Using Lemma 5 and Lemma 6, we can prove that $P^-(g,t)$ is the piecewise linear function given in the lemma. □

Lemma 16 *Let G be a go-theory, o be an object, and $\pi = \langle \Gamma, \sqsubseteq \rangle$ be a plan for G^o. Suppose $\gamma_1 \sqsubseteq \gamma_2 \ldots \sqsubseteq \gamma_n$ are the movements in π with $TCI(G^o, \pi, \gamma_k) = [T_k^-, T_k^+]$. $TCI(G^o, \pi, \gamma_i) \cup TCI(G^o, \pi, \gamma_{i+1}) \ldots \cup TCI(G^o, \pi, \gamma_j)$ is a* single time interval *iff for every $1 \leq i \leq k < j \leq n$ the following are true*

(i) $T_k^+ = T_{k+1}^-$

(ii) $loc_2(\gamma_k) = loc_1(\gamma_{k+1})$

Proof The lemma has two parts.

Only if part: If (i) and (ii) hold then $TCI(G^o, \pi, \gamma_i) \cup TCI(G^o, \pi, \gamma_{i+1}) \ldots \cup TCI(G^o, \pi, \gamma_j)$ is a single time interval. This result follows trivially from (i)

If part: If $TCI(G^o, \pi, \gamma_i) \cup TCI(G^o, \pi, \gamma_{i+1}) \ldots \cup TCI(G^o, \pi, \gamma_j)$ is a single time interval then (i) and (ii) are true.

Assume (i) is not true. Then for some k, either of the following should hold:

- $T_k^+ < T_{k+1}^-$

  This case is not possible because it implies there is a gap between T_k^+ and T_{k+1}^- which contradicts the assumption that $[T_i^-, T_j^+]$ is a single interval.

- $T_k^+ > T_{k+1}^-$

  This case is not possible either. It follows from Corollary A-1 that if for some γ $t \in TCI(G^o, \pi, \gamma)$ then every instance of π, satisfies γ w.r.t. π over a time interval that includes t. If $T_k^+ > T_{k+1}^-$ then in some instance of π both γ_k and γ_{k+1} has to be satisfied over intersecting time intervals which is contradicts the assumption that $\gamma_k \sqsubseteq \gamma_{k+1}$.

Now suppose (ii) is false. Then for some k, $loc_2(\gamma_k) \neq loc_1(\gamma_{k+1})$. We have already shown that (i) has to be true. Thus $T_k^+ = T_{k+1}^-$.

1. $T_k^+ = T^-(G^o, pi, \gamma_k, loc_2(\gamma_k))$ and $T_{k+1}^- = T^+(G^o, pi, \gamma_{k+1}, loc_1(\gamma_{k+1}))$;

   By definition of temporal certainty interval (Defintion 24).

2. By definition of earliest and latest arrival times; for every instance of π, say $\mathcal{I}$,

   (a) $\mathcal{I}$ satisfies γ_k w.r.t. π over a time interval $[t_1^k, t_2^k]$ such that $t_2^k \geq T_k^+$.

   (b) $\mathcal{I}$ satisfies γ_{k+1} w.r.t. π over a time interval $[t_1^{k+1}, t_2^{k+1}]$ such that $t_1^{k+1} \leq T_{k+1}^1$.

3. $t_2^k = T_k^+$

   Assume $t_2^k > T_k^+$ then by (2b) and $T_k^+ = T_{k+1}^-$, it follows that $t_1^{k+1} > t_2^k$ which by Definition 21 contradicts the assumption in (2) that I is an instance of π. Hence T_k^+ is the only valid value for t_2^k.

4. $t_1^{k+1} = T_{k+1}^-$

   Assume $t_1^{k+1} < T_{k+1}^-$ then by (2a) and $T_k^+ = T_{k+1}^-$, it follows that $t_1^{k+1} > t_2^k$ which by definition 21 contradicts the assumption in (2) that I is an instance of π.

5. $\mathcal{I}(o, T_k^-) = loc_2(\gamma_k)$

   By 3 and Definition 9 (satisfaction of a movement).

6. $\mathcal{I}(o, T_{k+1}^-) = loc_1(\gamma_{k+1})$

   By 4 and Definition 9 (satisfaction of a movement).

7. By defintion of a LOM interpretation $\mathcal{I}$ is a function thus $loc_2(\gamma_k) = loc_1(\gamma_{k+1})$ must hold when $T_k^+ = T_{k+1}^-$.

   Thus such a k can not exist and for all k, $loc_2(\gamma_k) = loc_1(\gamma_{k+1})$.

   □

Lemma 17 *Let G be a go-theory, o be an object, and π be a plan for G^o. Suppose γ is a movement in π such that $TCI(G^o, \pi, \gamma)$ is defined and $CheckPoints(\gamma) = [p_1, p_2 \ldots p_n]$. Then for every time point t in $TCI(G^o, \pi, \gamma)$, $P^+(G^o, \pi, \gamma, t)$ satisfies the following:*

$$P^{+}(G^{o},\pi,\gamma,t) = \begin{cases} \ell_1^{-}(t) & \textit{if}\, T_1^{-} \leq t \leq T_2^{-} \wedge t < t_1^{*} \\ \ell_1^{+}(t) & \textit{if}\, T_1^{-} \leq t \leq T_2^{-} \wedge t \geq t_1^{*} \\ \vdots & \\ \ell_i^{-}(t) & \textit{if}\, T_i^{-} \leq t \leq T_{i+1}^{-} \wedge t < t_i^{*} \\ \ell_i^{+}(t) & \textit{if}\, T_i^{-} \leq t \leq T_{i+1}^{-} \wedge t \geq t_i^{*} \\ \vdots & \\ \ell_{n-1}^{-}(t) & \textit{if}\, T_{n-1}^{-} \leq t \leq T_n^{-} \wedge t < t_{n-1}^{*} \\ \ell_{n-1}^{+}(t) & \textit{if}\, T_{n-1}^{-} \leq t \leq T_n^{-} \wedge t \geq t_{n-1}^{*} \end{cases}$$

where

- $T_i^{-} = T^{-}(G^{o},\pi,\gamma,p_i)$
- $\ell_i^{-}(t) = p_i + \vec{\gamma}\,[(t - T_i^{-}) \times v^{+}(\gamma,p_i,p_{i+1})]$
- $\ell_i^{+}(t) = p_{i+1} + \vec{\gamma}\,[(t - T_{i+1}^{-}) \times v^{-}(\gamma,p_i,p_{i+1})]$
- $t_i^{*} \in \mathbf{R}$ *such that* $\ell_i^{-}(t_i^{*}) = \ell_i^{+}(t_i^{*})$

Similarly $P^{-}(G^{o},\pi,\gamma,t)$ *is piecewise linear.*

Proof The proof is very similar to proof of Lemma 4 which constructs an instance of π such that for any two consecutive check points p_i and p_{i+1}, $\mathcal{I}(o,T^{-}(G^{o},\pi,\gamma,p_i)) = p_i$ and $\mathcal{I}(o,T^{-}(G^{o},\pi,\gamma,p_{i+1})) = p_{i+1}$. For this proof we need to show that we can construct a model which arrives at every check point at the earliest possible arrival time. We are going to iteratively construct such a model.

For readability let's assume $T^{-}(G^o, \pi, \gamma, p_i) = T_i^-$, $T^{-}(G^o, \pi, \gamma, p_{i+1}) = T_{i+1}^-$, $v^-(\gamma, p_i, p_{i+1}) = v_i^-$ and $v^+(\gamma, p_i, p_{i+1}) = v_i^+$.

Basic Step Let $\mathcal{I}_0$ and $\mathcal{I}_{12}$ be interpretations such that

- $\mathcal{I}_0(o, T_1^-) = p_1, \mathcal{I}_0(o, t_2) = p_2$ and $\mathcal{I}_0$ satisfies γ w.r.t. π oven an interval including T_1^- and t_{i+1}.
- $\mathcal{I}_{12}(o, t_1) = p_1, \mathcal{I}_{12}(o, T_2^-) = p_2$ and $\mathcal{I}_{12}$ satisfies γ w.r.t. π oven an interval including t_i and T_2^-.

Using $\mathcal{I}_0$ and $\mathcal{I}_{12}$ define a LOM interpretation $\mathcal{I}_1$ as follows:

$$\mathcal{I}_1(o,t) = \begin{cases} \mathcal{I}_0(o,t) & \text{if } t \le T_1^- \\ p_1 + v_1^+(t - T_1^-)\vec{\gamma} & \text{if } T_1^- < t < T_2^- \wedge t < t^* \\ p_2 - v_1^-(T_2^- - t)\vec{\gamma} & \text{if } T_1^- < t < T_2^- \wedge t^* \le t \\ \mathcal{I}_{12}(o,t) & \text{if } t \ge T_2^- \end{cases}$$

where $t^* \in \mathbf{R}$ such that $p_i + v_1^+(t^* - T_1^-)\vec{\gamma} = p_{i+1} - v_1^-(T_2 - t^*)\vec{\gamma}$

As shown in Lemma 4, $\mathcal{I}_1$ is an instance of π and for every $P \in [p_1, p_2]$, we have $\mathcal{I}(o, T^-(G^o, \pi, \gamma, P))) = P$.

Iterative Step For every consecutive check points p_i, p_{i+1} such that $2 \le i < n$ in order, construct $\mathcal{I}_i$ using $\mathcal{I}_{i-1}$ and any model $\mathcal{I}_{i2}$, such that $\mathcal{I}_{i2}(o, t_i) = p_i, \mathcal{I}_{i2}(o, T_{i+1}^-) = p_{i+1}$ and $\mathcal{I}_{i2}$ satisfies γ w.r.t. π oven an interval including t_i and T_{i+1}^-. Using $\mathcal{I}_{i-1}$ and $\mathcal{I}_{i2}$ define a LOM interpretation $\mathcal{I}_i$ as follows:

$$\mathcal{I}_i(o,t) = \begin{cases} \mathcal{I}_{i-1}(o,t) & \text{if } t \leq T_i^- \\ p_i + v_i^+(t - T_i^-)\vec{\gamma} & \text{if } T_i^- < t < T_{i+1}^- \wedge t < t_i^* \\ p_{i+1} - v_i^-(T_{i+1}^- - t)\vec{\gamma} & \text{if } T_i^- < t < T_{i+1}^- \wedge t_i^* \leq t \\ \mathcal{I}_{i2}(o,t) & \text{if } t \geq T_{i+1}^- \end{cases}$$

where $t_i^* \in \mathbf{R}$ such that $p_i + v_i^+(t_i^* - T_i^-)\vec{\gamma} = p_{i+1} - v_i^-(T_{i+1}^- - t_i^*)\vec{\gamma}$

By induction and proof of Lemma 4 we can show that $\mathcal{I}_{n-1}$ is an instance of π and furthermore for every $P \in LS(\gamma)$, $\mathcal{I}_{n-1}(o, T^-(G^o, \pi, \gamma, P))) = P$. It follows from Lemma 6, that $T^-(G^o, \pi, \gamma, P)$ and $P^+(G^o, \pi, \gamma, t)$ are inverse functions hence $P^+(G^o, \pi, \gamma, t) = \mathcal{I}_{n-1}(o,t)$ which is by construction equal to the piecewise linear function in the lemma throughout $[T_1^-, T_n^-]$. $\square$

Theorem 9 *Let G be a go-theory, o, o' be objects, π, π' be plans for G^o and $G^{o'}$ respectively and $b = \mathsf{near}(o, o', d, t_1, t_2)$ be a ground atom. All instances of π and π satisfy b iff all the following conditions hold:*

1. *There is a subset S of the set of movements in π such that S is temporally relevant to $[t_1, t_2]$*

2. *There is a subset S' of the set of movements in π' such that S' is temporally relevant to $[t_1, t_2]$*

3. *$\forall t \in [t_1, t_2]\ \exists \gamma \in S \wedge \exists \gamma' \in S'$ such that $t \in TCI(G^o, \pi, \gamma)$ and $t \in TCI(G^{o'}, \pi', \gamma')$ and*

   a) $\mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)) \leq d$ *and*

   b) $\mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)) \leq d$ *and* *c)* $\mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^-(G^{o\prime}, \pi', \gamma', t)) \leq d$ *and*

d) $\mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^+(G^{o\prime}, \pi', \gamma', t)) \leq d$ *and*

Proof The theorem has two parts.

If part: If all instances of π and π' satisfy b then all of the conditions hold. Assume condition (1) is false. Then there is at least one time point $t \in [t_1, t_2]$ such that t is not in the temporal certainty interval o for any movement in π. Since the location of the object is only bounded for the time points in temporal certainty interval of a movement, there can be an instance where o is more than d apart form o', which does not satisfy the atom b. This contradicts the initial assumption. Same reasoning proves that condition (2) must hold. By conditions 1 and 2 every time point in $[t_1, t_2]$ is also in the temporal certainty interval of some movement in π and π'. Hence in all instances of π and π' the locations of both objects are restricted within the positional certainty region of the movements that are being satisfied. Then as shown in Lemma 14, the maximum distance between the objects can not be more than the distance on the end points of the positional certainty region. Since in all instances of π and π', b is satisfied then the distance between two objects is always less than or equal to d. Thus the conditions (3a) to (3d) must hold.

Only if part: If all of the conditions hold then all instances of π and π' satisfy b. This part is trivial. By conditions 1 and 2 every time point in $[t_1, t_2]$ is also in the temporal certainty interval of some movement in π and π'. Furthermore by Lemma 16, for every $t \in [t_1, t_2]$ this movement is either unique or t is at the meeting point of temporal certainty intervals of two movements (in which case in all instances $I(o, t)$ is equal to the same point, so it does not matter which movement is chosen). Hence in all instances of

π and π' the locations of both objects are restricted within the positional certainty region of the movements that are being satisfied. Then as shown in Lemma 14, the maximum distance between the objects can not be more than the distance on the end points of the positional certainty region. Since the conditions (3a) to (3d) hold then in all instances of π and π' the distance between two objects is always less than or equal to d. Thus all instances of π and π' satisfy the atom b. $\Box$

Theorem 10 *Suppose G is a consistent* go*-theory and $b = \mathsf{near}(o, o', d, t_1, t_2)$ is a ground atom. b is a logical consequence of G iff for every plan π for G^o, and π' of $G^{o'}$ algorithm* $\mathsf{CheckNear}(G, \pi, \pi', b)$ *returns "true".*

Proof It is easy to see that Algorithm $\mathsf{CheckNear}(G, \pi, \pi', b)$ returns true only when the conditions in Lemma Theorem 9 are all true.

If part: If $G \models a$ then for every plan π of G^o,π' of $G^{o'}$ $\mathsf{CheckNear}(G, \pi, \pi', b)$ returns "true". Assume there is a pair of plans π for G^o and π' for $G^{o'}$ such that algorithm returns false. Then by Theorem 9 there is an interpretation $\mathcal{I}$ that is an instance of π and π' and $I \not\models b$. By definition of instance (Definition 21), $I \models G^o$ and $\mathcal{I} \models G^{o'}$. Since G is consistent there is a model of G, say $\mathcal{I}'$. We construct another model $\mathcal{I}^*$ of G as follows:

- $\forall t \, \forall \, obj, \; \mathcal{I}^*(obj, t) = I(obj, t)$ when $obj = o$ or $obj = o'$
- $\forall t \, \forall \, obj, \; \mathcal{I}^*(obj, t) = I'(obj, t)$ otherwise.

It is trivial to see that $\mathcal{I}^* \models G$ and $\mathcal{I}^* \not\models b$. But this contradicts $G \models b$.

Only if part: If for every pair of plans π for G^o and π' for $G^{o'}$, the algorithm $\mathsf{CheckNear}(G, \pi, \pi', b)$ returns "true" then $G \models b$. Assume $G \not\models b$. Then there is a model of G, $\mathcal{I}^*$ such that $\mathcal{I}^* \not\models b$. As shown in Theorem 2, for every model $\mathcal{I}'$ of G there is a plan π' such that $\mathcal{I}'$ is an instance of π'. Let $\mathcal{I}_o$ and $\mathcal{I}'_o$ be the restrictions of $\mathcal{I}^*$ to o and o' respectively. Assume $\mathcal{I}_o$ is an instance of π^* and $\mathcal{I}_o$ is an instance of π'. Then by Theorem 9, $\mathsf{CheckNear}(G, \pi, \pi', b)$ returns false but this contradicts the initial assumption of for every pair of plans the algorithm returns "true". Hence $\mathcal{I}^*$ can not exist.

$\square$

Ground atomic $\mathsf{far}()$ queries

Corollary A-4 *Let g be a go- atom such that $TCI(g)$ is defined. If $t \in TCI(g)$, then for every point $P \in PCR(g,t)$ there is a model $\mathcal{I}$ of G such that $\mathcal{I}(o,t) = P$.*

Proof Suppose $P_1 = loc_1(g)$ and $P_2 = loc_2(g)$. Also let $S^- = T^-(g, P_1)$ and $S^+ = T^+(g, P_1)$ and $E^- = T^-(g, P_2)$ and $E^+ = T^+(g, P_2)$. By definition of satisfaction of a go atom, it is enough to show the following two conditions hold for any point $P \in PCR(g,t)$.

(1) $[t - d1/v^-(g), t - d1/v^+(g)] \cap [S^-, S^+] \neq \emptyset$

(2) $[d2/v^+(g) + t, d2/v^-(g) + t] \cap [E^-, E^+] \neq \emptyset$

where $d1 = \mathsf{dist}(P, P_1)$ and $d2 = \mathsf{dist}(P, P_2)$.

Let $P^- = P^-(g,t)$ and $P^+ = P^+(g,t)$. By definition of minimum/maximum advancement point there are models $\mathcal{I}_1$ and $\mathcal{I}_2$ of G such that $\mathcal{I}_1(o,t) = P^-$ and $\mathcal{I}_2(o,t) = P^+$. The following four conditions hold:

(i) $[t - d1^-/v^-(g), t - d1^-/v^+(g)] \cap [S^-, S^+] \neq \emptyset$

(ii) $[d2^-/v^+(g) + t, d2^-/v^-(g) + t] \cap [E^-, E^+] \neq \emptyset$

(iii) $[t - d1^+/v^-(g), t - d1^+/v^+(g)] \cap [S^-, S^+] \neq \emptyset$

(iv) $[d2^+/v^+(g) + t, d2^+/v^-(g) + t] \cap [E^-, E^+] \neq \emptyset$

where $d1^- = \mathsf{dist}(P^-, P_1)$, $d2^- = \mathsf{dist}(P^-, P_2)$ and $d1^+ = \mathsf{dist}(P^+, P_1)$, $d2^+ = \mathsf{dist}(P^+, P_2)$.

Since $P \in [P^-, P^+]$; $d1^- \leq d1 \leq d1^+$. Thus it follows from (i) and (iii) that (1) holds.

Since $P \in [P^-, P^+]$; $d2^+ \leq d2 \leq d2^-$. Thus it follows from (ii) and (iv) that (2) holds.

□

Lemma 18 *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*

(1) $t \in TCI(g)$ and $t \in TCI(g')$ and

(2) The minimum distance between line segments $PCR(g, t)$ and $PCR(g', t)$ is greater than d.

Proof The lemma has two parts.

Only if part: Show that when these conditions hold $G \models \mathsf{far}(o, o', d, t, t)$. If $t \in TCI(g)$ then by Lemma 7, for every model $\mathcal{I}$ of G, $\mathcal{I}(o, t)$ is on the line segment $PCR(g, t)$. Similarly if $t \in TCI(g')$ then for every model $\mathcal{I}$ of G, $\mathcal{I}(o', t)$ is on the line segment $PCR(g', t)$. If the minimum distance between any point on $PCR(g, t)$ and $PCR(g', t)$ is more than d then for every model $\mathcal{I}$ of G, $\mathsf{dist}(\mathcal{I}(o, t), \mathcal{I}(o', t)) > d$.Thus every model of G also satisfies $\mathsf{far}(o, o', d, t, t)$

If part: Show that if $G \models \mathsf{far}(o, o', d, t, t)$ then all the conditions are satisfied. Assume $TCI(g)$ is not defined. Then by Definition 24, $T^+(g, loc_1(g)) \geq T^-(g, loc_2(g))$.

By definition of earliest and latest arrival times there is an interpretation $\mathcal{I}_1$ that satisfy g over $[T^+(g, loc_1(g)), t_2]$, for some t_2. Similarly there is an interpretation $\mathcal{I}_2$ that satisfy g over $[t_1, T^-(g_1, loc_2(g))]$, for some t_1. If $t < T^+(g, loc_1(g))$ then $I_1(o, t)$ can be any point that is less than d distant to $\mathcal{I}_1(o', t)$. If $t > T^-(g, loc_2(g))$ then $I_2(o, t)$ can be any point that is less than d distant to $\mathcal{I}_2(o', t)$.Thus either $\mathcal{I}_1 \not\models \mathsf{far}(o, o', d, t, t)$ or $\mathcal{I}_2 \not\models \mathsf{far}(o, o', d, t, t)$, which contradicts the assumption that $G \models \mathsf{far}(o, o', d, t, t)$.

As a result of similar reasoning $TCI(g')$ should be defined and (1) has to hold.

Assume (2) is false. By (1) and corollary A-4 for every point P on $PCR(g, t)$ there is a model $\mathcal{I}$ of G such that $\mathcal{I}(o, t) = P$. Let P and Q be two points on $PCR(g, t)$ and $PCR(g', t)$ respectively such that $\mathsf{dist}(P, Q) \leq d$ (this is possible because we assumed (2) is false). Then there is a model $\mathcal{I}$ such that $\mathcal{I}(o, t) = P$ and $\mathcal{I}(o', t) = Q$ hence $\mathcal{I} \not\models \mathsf{far}(o, o', d, t, t)$, which contradicts the assumption that $G \models \mathsf{far}(o, o', d, t, t)$. $\square$

Lemma 21 *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*

(1) $[t_1, t_2] \subseteq TCI(g)$ *and* $[t_1, t_2] \subseteq TCI(g')$

(2) $Nbr(g, [t_1, t_2], d) \cap SE(g', [t_1, t_2]) = \emptyset$

Proof The lemma has two parts.

Only if part: If all the conditions are satisfied then $G \models f$.

Assume $G \not\models f$ then

(i) There is a $t \in [t_1, t_2]$ and a model $\mathcal{I}$ of G such that $\mathsf{dist}(\mathcal{I}(o, t), \mathcal{I}(o', t)) \leq d$.

(ii) By condition (1) and Lemma 7; $\mathcal{I}(o, t) \in PCR(g, t)$ and $\mathcal{I}(o', t) \in PCR(g', t)$.

(iii) By definition of d-neighborhood and (i); $I(o', t) \in Nbr(g, [t, t], d)$.

Thus $(\mathcal{I}(o', t), t) \in Nbr(g, [t, t], d) \cap SE(g', [t, t])$.

(iv) By definition of d-neighborhood and space envelope;

$Nbr(g, [t, t], d) \subseteq Nbr(g, [t_1, t_2], d)$ and $SE(g', [t, t], d) \subseteq SE(g, [t_1, t_2], d)$.

(v) By (iv) and (iii); $Nbr(g, [t_1, t_2], d) \cap SE(g', [t_1, t_2]) \neq \emptyset$, which is a contradiction.

If part: If $G \models f$ then all the conditions are satisfied.

Assume (1) is not true leads to a contradiction. The proof is very similar to the second part of the proof for Lemma 18. Assume (2) is not true. Then there is a point P and a time point $t \in [t_1, t_2]$ such that $(P, t) \in Nbr(g, [t_1, t_2], d) \cap SE(g', [t_1, t_2])$. By definition of space envelope and d-neighborhood there is a model of G, say $\mathcal{I}$ such that $\mathcal{I}(o', t) = P$ and $\mathcal{I}(o, t) = Q$ where $\mathsf{dist}(P, Q) \leq d$. Thus $\mathcal{I} \not\models f$ which is a contradiction.

□

Theorem 11 *Suppose G is a go-theory and o, o' are objects.Let π and π' be plans of G^o and $G^{o'}$. The ground atom $f = \mathsf{far}(o, o', d, t_1, t_2)$ is satisfied in all instances of π and π' iff all the following hold:*

(1) $\exists S \subseteq \pi$ such that S is temporally relevant to $[t_1, t_2]$

(2) $\exists S' \subseteq \pi'$ such that S' is temporally relevant to $[t_1, t_2]$

(3) $Nbr(G^o, \pi, [t_1, t_2], d) \cap SE(G^{o'}, \pi', [t_1, t_2]) = \emptyset$.

Proof The theorem has two parts.

If part: If all instances of π and π' satisfy f then all of the conditions hold. Assume condition (1) is false. Then there is at least one time point $t \in [t_1, t_2]$ such that t is not in the temporal certainty interval o for any movement in π. Since the location of the object is only bounded for the time points in temporal certainty interval of a movement, there can be an instance where o is less than d apart form o', which does not satisfy the atom f. This contradicts the initial assumption. Same reasoning proves that condition (2) must hold. By conditions 1 and 2 every time point in $[t_1, t_2]$ is also in the temporal certainty interval of some movement in π and π'. Hence in all instances of π and π' the locations of both objects are restricted within the positional certainty region of the movements that are being satisfied. Then as shown in the proof of Lemma 21, $Nbr(G^o, \pi, [t_1, t_2], d) \cap SE(G^{o'}, \pi', [t_1, t_2]) = \emptyset$.

Only if part: If all of the conditions hold then all instances of π and π' satisfy f. This part is trivially follows from Lemma 21. $\square$

Theorem 12 *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 plan π and π' of G^o and $G^{o'}$, the algorithm* $\mathsf{CheckFar}$*(G, π, π', f) returns "true".*

Proof It is easy to see that Algorithm $\mathsf{CheckFar}(G, \pi, \pi', f)$ returns true only when the conditions in Theorem 11 are all true.

If part: If $G \models f$ then for every plan π of G^o,π' of $G^{o\prime}$ $\mathsf{CheckFar}(G, \pi, \pi', f)$ returns "true". Assume there is a pair of plans π for G^o and π' for $G^{o\prime}$ such that algorithm returns false. Then by Theorem 11 there is an interpretation $\mathcal{I}$ that is an instance of π and π' and $I \not\models f$. By definition of instance (Definition 21), $\mathcal{I} \models G^o$ and $\mathcal{I} \models G^{o\prime}$. Since G is consistent there is a model of G, say $\mathcal{I}'$. We construct another model $\mathcal{I}^*$ of G as follows:

- $\forall t\, \forall\, obj,\ \mathcal{I}^*(obj, t) = I(obj, t)$ when $obj = o$ or $obj = o'$

- $\forall t \, \forall \, obj, \; \mathcal{I}^*(obj, t) = I'(obj, t)$ otherwise.

It is trivial to see that $\mathcal{I}^* \models G$ and $\mathcal{I}^* \not\models f$. But this contradicts $G \models f$.

Only if part: If for every pair of plans π for G^o and π' for $G^{o\prime}$, $\mathsf{CheckFar}(G, \pi, \pi', f)$ returns “true” then $G \models b$. Assume $G \not\models f$. Then there is a model of G, $\mathcal{I}^*$ such that $\mathcal{I}^* \not\models f$. As shown in Theorem 2, for every model $\mathcal{I}'$ of G there is a plan π' such that $\mathcal{I}'$ is an instance of π'. Let $\mathcal{I}_o$ and $\mathcal{I}'_o$ be the restrictions of $\mathcal{I}^*$ to o and o' respectively. Assume $\mathcal{I}_o$ is an instance of π^* and $\mathcal{I}_o$ is an instance of π'. Then by Theorem 11, $\mathsf{CheckFar}(G, \pi, \pi', f)$ returns false but this contradicts the initial assumption of for every pair of plans the algorithm returns “true”. Hence $\mathcal{I}^*$ can not exist. $\Box$

Answering Non-Ground Atomic Queries

go Queries

Lemma 22 *Let G be a go theory, $q = \mathsf{go}(o, P_1, P_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a normalized ground go atom, and $Q = \mathsf{go}(o, P_1, P_2, T_1^-, T_1^+, T_2^-, T_2^+, V^-, V^+)$ be a ground go atom such that:*

$$T_1{}^- \le t_1{}^- \le t_1{}^+ \le T_1{}^+ \textit{ and } T_2{}^- \le t_2{}^- \le t_2{}^+ \le T_2{}^+ \textit{ and } V^- \le v^- \le v^+ \le V^+$$

If $G \models q$ then $G \models Q$.

Proof If q is a logical consequence of G then by definition $\forall \mathcal{I} \models G, \; \mathcal{I} \models q$. To prove the lemma it is sufficient to show that $\forall \mathcal{I} \models q, \; \mathcal{I} \models Q$ as the rest will follow from the definition of logical consequence. Let $\mathcal{I}$ be a LOM interpretation that satisfies q then by definition the following are true;

1. $\exists\, t_1 \in [t_1^-, t_1^+]$ such that $\mathcal{I}(o, t_1) = P_1$,

2. $\exists\, t_2 \in [t_2^-, t_2^+]$ such that $\mathcal{I}(o, t_2) = P_2$

3. $\mathcal{I}(o, t)$ maps the interval $[t_1, t_2]$ one-to-one onto the line segment $[P_1, P_2]$.

4. At all but finitely many points in $[t_1, t_2]$, the derivative $v(t) = d(|\mathcal{I}(o, t)|)/dt$ (which represents o's speed) is defined, and $v^- \leq v(t) \leq v^+$.

   Moreover $\mathcal{I}$ also satisfies the following ;

i. $\exists\, t_1 \in [T_1^-, T_1^+]$ such that $\mathcal{I}(o, t_1) = P_1$, by 1 and a

ii. $\exists\, t_2 \in [T_2^-, T_2^+]$ such that $\mathcal{I}(o, t_2) = P_2$, by 2 and b

iii. $I(o, t)$ maps the interval $[t_1, t_2]$ one-to-one onto the line segment $[P_1, P_2]$, by 3.

iv. At all but finitely many points in $[t_1, t_2]$, the derivative $v(t) = d(|\mathcal{I}(o, t)|)/dt$ (which represents o's speed) is defined, and $v^- \leq v(t) \leq v^+$, by 4 and c.

Hence $\mathcal{I}$ satisfies Q as well. Thus Q is a logical consequence of G if q is a logical consequence of G. $\square$

Theorem 13 *Let G be consistent simple go theory and $q = \mathsf{go}(o, p_1, p_2, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+)$ be a go atom. Suppose π is the plan for G^o. There is a solution to q w.r.t. G iff there is a movement γ in π such that $\Theta_{q,G^o,\gamma}$ is satisfiable. Furthermore $\Theta_{q,G^o,\gamma}$ is a solution to q w.r.t. G.*

Proof The theorem has two parts.

If part If there is a solution to q w.r.t. G then there is a movement γ in π such that $\Theta_{q,G^o,\gamma}$

is solvable. Without losing any generality we will consider a solution to q w.r.t. G that has only one satisfying substitution ϕ. Then by lemma 10, the following conditions hold:

(i) There is a movement γ in π that is relevant to $\phi(q)$

(ii) $\phi[t_1^-] \leq T^-(G^o, \pi, \gamma, \phi[p_1]) \leq T^+(G^o, \pi, \gamma, \phi[p_1]) \leq \phi[t_1^+]$.

(iii) $\phi[t_2^]- \leq T^-(G^o, \pi, \gamma, \phi[p_2]) \leq T^+(G^o, \pi, \gamma, \phi[p_2]) \leq \phi[t_2^+]$

(iv) $\phi[v^-] \leq V^-(\gamma, P_1, P_2) \leq V^+(\gamma, P_1, P_2) \leq \phi[v^+]$.

where $\phi[X]$ is the value assigned to variable X in ϕ. If X is a constant then $\phi[X]$ is X.

By condition (i) we know that there is a movement γ that is relevant to $\phi(q)$. Let $P_1 = loc_1(\gamma)$ and $P_1 = loc_2(\gamma)$. Then condition (i) translates to the following three conditions:

-0 $\leq d_1 \leq d_2 \leq \mathsf{dist}(P_1, P_2)$

-$\phi[p_1] = P_1 + d_1 * \vec{\gamma}$

-$\phi[p_2] = P_1 + d_2 * \vec{\gamma}$

These constraints mark the line segment on $LS(\gamma)$ that is also in $LS(\phi(q))$. Note that $\Theta_{q,G^o,\gamma}$ also contains these constraints and the conditions (ii) to (iv) with a replacement of every $\phi[X]$ to X. Thus ϕ also satisfies $\Theta_{q,G^o,\gamma}$.

Only if part If there is a movement γ in π such that $\Theta_{q,G^o,\gamma}$ is solvable then there is a solution to to q w.r.t. G. For this part we are going to show that $\Theta_{q,G^o,\gamma}$ is a solution to q w.r.t. G . Let ϕ be a substitution that satisfies $\Theta_{q,G^o,\gamma}$ such that (d_1, v_1) and (d_2, v_2) is in ϕ for some values v_1, v_2. Given v_1, v_2, the constraints in $\Theta_{q,G^o,\gamma}$ basically selects a subsegment of $LS(\gamma)$. Furthermore the earliest/latest arrival times to these points set the minum and maximum values for temporal variables. Note that the temporal constraints in

$\Theta_{q,G^o,\gamma}$ (i.e. the constraints related to t_1 and t_2) is same as in Lemma 10 which are proven to be necessary and sufficient for a ground go atom entailment. Thus for any substitution ϕ that satisfy $\Theta_{q,G^o,\gamma}$, $G \models \phi(q)$. Hence $\Theta_{q,G^o,\gamma}$ is a solution to q w.r.t. G. $\Box$

$\mathsf{in}()$ Queries

Lemma 23 *Let G be a go theory, $q = \mathsf{in}(o, p_1, p_2, t_1, t_2)$ and $Q = \mathsf{in}(o, P_1, P_2, t_1, t_2)$ be ground atoms such that:*

(i) ${P_1}^x \leq {p_1}^x$ *and* ${P_1}^y \leq {p_1}^y$ *and* ${P_1}^z \leq {p_1}^z$

(ii) ${P_2}^x \geq {p_2}^x$ *and* ${P_2}^y \geq {p_2}^y$ *and* ${P_2}^z \geq {p_2}^z$

If $G \models q$ then $G \models Q$.

Proof If q is a logical consequence of G then by definiton $\forall \mathcal{I} \models G,\ \mathcal{I} \models q$. To prove the lemma it is sufficient to show that $\forall \mathcal{I} \models q,\ \mathcal{I} \models Q$ as the rest will follow from the definition of logical consequence. Let $\mathcal{I}$ be a LOM interpretation that satisfies q then by definition of satisfaction of an in atom there is a time t and a point P such that ;

1. $\mathcal{I}(o, t) = P$ and
2. $t \in [t_1, t_2]$ and
3. $P^x \in [{p_1}^x, {p_2}^x]$, and $P^y \in [{p_1}^y, {p_2}^y]$ and $P^z \in [{p_1}^z, {p_2}^z]$ such that $I(o, t) = (x, y, z)$.

   Moreover by (3), (i) and (ii); the following is also true ;

(4) $P^x \in [{P_1}^x, {P_2}^x]$, and $P^y \in [{P_1}^y, {P_2}^y]$ and $P^z \in [{P_1}^z, {P_2}^z]$

Hence by definition of satisfaction of an in atom with (1), (2) and (4) $\mathcal{I}$ satisfies Q as well. Thus Q is a logical consequence of G if q is a logical consequence of G. $\square$

Lemma 24 *Let G be a go theory, $q = \mathsf{in}(o, p_1, p_2, t_1, t_2)$ and $Q = \mathsf{in}(o, p_1, p_2, T_1, T_2)$ be ground atoms such that $T_1 \leq t_1$ and $T_2 \geq t_2$. If $G \models q$ then $G \models Q$.*

Proof If q is a logical consequence of G then by definiton $\forall \mathcal{I} \models G,\ \mathcal{I} \models q$. To prove the lemma it is sufficient to show that $\forall \mathcal{I} \models q,\ \mathcal{I} \models Q$ as the rest will follow from the definition of logical consequence. Let $\mathcal{I}$ be a LOM interpretation that satisfies q then by definition of satisfaction of an in atom there is a time t and a point P such that ;

1. $\mathcal{I}(o, t) = P$ and
2. $t \in [t_1, t_2]$ and
3. $P^x \in [{p_1}^x, {p_2}^x]$, and $P^y \in [{p_1}^y, {p_2}^y]$ and $P^z \in [{p_1}^z, {p_2}^z]$ such that $I(o, t) = (x, y, z)$.

Moreover since $T_1 \leq t_1$ and $T_2 \geq t_2$ the following is also true:

(4) $t \in [T_1, T_2]$

Hence by definition of satisfaction of an in atom with (1), (3) and (4) $\mathcal{I}$ satisfies Q as well. Thus Q is a logical consequence of G if q is a logical consequence of G. $\square$

Theorem 14 *Let G be consistent simple go theory and let $q = \mathsf{in}(o, P, Q, t_1, t_2)$ be an atom. Suppose π is the main plan for G^o. There is a solution to to q w.r.t. G iff there is a movement γ in π such that $\Omega_{q,G^o,\gamma}$ is solvable (i.e. there exists a substitution that satisfies the constraints in Ω).*

Proof The theorem has two parts.

If part If there is a solution to q w.r.t. G then there is a movement γ in π such that $\Omega_{q,G^o,\gamma}$ is solvable. Without losing any generality we will consider a solution to q w.r.t. G that has only one satisfying substitution ϕ. Then by lemma 12, the following conditions hold:

(i) There is a movement γ in π that is related to $\phi(a)$

(ii) $T^+(G^o, \pi, \gamma, p_1) \leq \phi[t_2]$

(iii) $T^-(G^o, \pi, \gamma, p_2) \geq \phi[t_1]$

where $[p_1, p_2] = LS(\gamma) \cap Vol(\phi(q))$ and $\phi[X]$ is the value assigned to X in ϕ when X is a variable. If X is a constant then $\phi[X]$ is X.

By condition (i) we know that there is a movement γ that is spatially related. Let $P_1 = loc_1(\gamma)$ and $P_1 = loc_2(\gamma)$. Then condition (i) translates to the following four conditions:

-$0 \leq d_1 \leq d_2 \leq \mathsf{dist}(P_1, P_2)$

-$p_1 = P_1 + d_1 * \vec{\gamma}$ and $p_2 = P_1 + d_2 * \vec{\gamma}$

-$P^x \leq min(p_1^x, p_2^x)$, $P^y \leq min(p_1^y, p_2^y)$, $P^z \leq min(p_1^z, p_2^z)$,

-$Q^x \geq max(p_1^x, p_2^x)$, $Q^y \geq max(p_1^y, p_2^y)$, $Q^z \geq max(p_1^z, p_2^z)$

The first two constraints set the line segment on $LS(\gamma)$ that is also in $Vol(\phi(q))$. Last two states that $Vol(\phi(q))$ is actually at least as big as to contain $[p_1, p_2]$. Note that $\Omega_{q,G^o,\gamma}$ also contains these constraints and the conditions (ii), (iii) with a replacement of $\phi[X]$ to X. Thus ϕ also satisfies $\Omega_{q,G^o,\gamma}$.

Only if part If there is a movement γ in π such that $\Omega_{q,G^o,\gamma}$ is solvable then there is a solution to to q w.r.t. G. For this part we are going to show that $\Omega_{q,G^o,\gamma}$ is a solution to q w.r.t. G . Let ϕ be a substitution that satisfies $\Omega_{q,G^o,\gamma}$ such that (d_1, v_1) and (d_2, v_2) is in ϕ for some values v_1, v_2. Given v_1, v_2, the constraints in $\Omega_{q,G^o,\gamma}$ basically selects a subsegment of $LS(\gamma)$ whose end points set the minimal volume. Furthermore the earliest/latest arrival times to these points set the minimal time window. The minimal volume ensures that the assigned query volume will contain the line segment. Note that the temporal constraints in $\Omega_{q,G^o,\gamma}$ (i.e. the constraints related to t_1 and t_2) is same as in Lemma 12 which are proven to be necessary and sufficient for a ground in atom entailment. Thus for any substitution ϕ that satisfy $\Omega_{q,G^o,\gamma}$, $G \models \phi(q)$. $\square$

$\mathsf{near}()$ Queries

Lemma 25 *Let G be a simple go theory and let o, o' be two objects. Then there are disjoint time intervals $T_1, T_2, \ldots, T_n$ and quadratic functions $f_1(t), f_2(t), \ldots f_n(t)$ such that $\Delta_G(o, o', t) = d$ iff $f_i(t) = d$ for some i such that $t \in T_i$.*

Proof When G is a simple go theory, we know that there is a main plan π for G^o and a main plan π' for $G^{o'}$. Suppose γ is a movement of π and γ' is a movement of π'. Then we can compute $\Delta_G(o, o', t)$ in the following manner:

- For every t in $TCI(G^o, \pi, \gamma)$ and $TCI(G^{o'}, \pi', \gamma')$

$$\begin{aligned}\Delta_G(o, o', t) = max(\ & \mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)),\\ & \mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)),\\ & \mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t)),\\ & \mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t)))\end{aligned}$$

- ∞, otherwise.

For any time point that is not in the temporal certainty interval of both objects for unique movements $\Delta_G(o, o', t)$ will be constant. For the other case obviously the maximum distance will be the euclidean distance between a pair of piecewise linear functions parametrized by time, hence $\Delta_G(o, o', t)$ will be quadratic w.r.t. time. For any subinterval T of the temporal certainty intervals the behavior of $\Delta_G(o, o', t)$ will be same as one of the four distance equations. We can break the common temporal certainty interval into disjoint subintervals such that during each subinterval the maximum of four distance equations is always the same equation. $\Box$

Lemma 26 *Let G be a simple go theory and o, o' be two objects. The total number of critical time points in $\Delta_G(o, o', t)$ is bounded by $O(n)$ where n is the total number of go atoms in G^o and $G^{o'}$.*

Proof Let γ and γ' be two movements in the main plans of G^o and $G^{o'}$ such that temporal certainty intervals of γ and γ' overlap. For any time point t in the common temporal certainty interval $Delta_G(o, o', t)$ is the maximum of one of the following:

f1) $\mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t))$

f2) $\mathsf{dist}(P^-(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t))$

f3) $\mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^-(G^{o'}, \pi', \gamma', t))$

f4) $\mathsf{dist}(P^+(G^o, \pi, \gamma, t), P^+(G^{o'}, \pi', \gamma', t))$

It follows from Lemma 17, in the most general case the functions P^- and P^+ have at most $2n$ line segments where n is the number of go atoms in the movement. Hence the functions (f1) to (f4) are composed of at most $4n$ quadratic functions where n is the maximum number of atoms in γ and γ'. $Delta_G(o, o', t)$ might change behavior when the following functions change sign: $f1-f2$, $f1-f3$, $f1-f4$, $f2-f3$, $f2-f4$ and $f3-f4$. It is clear to see that each function $fi-fj$ is composed of at most $8n$ pieces. Sign change can happen at the roots of each piece which are quadratic. Thus for any $fi-fj$ there are at most $16n$ time points where function might change its sign. Since we have 6 of these functions the total number of critical time points will be bounded by $96n$. (Note that this is a very relaxed bound and not every sign change would trigger a behavior change for $Delta_G(o, o', t)$.) Finally we generalize this calculation trivially for the entire theory. $\square$

Lemma 27 *Let G be a* go *theory and $q = near(o, o', D, T_1, T_2)$ be a ground near atom such that $G \models q$. Then the following are true:*

- *(i)$G \models near(o, o', d, T_1, T_2)$ where $D \leq d$.*
- *(ii) $G \models near(o, o', D, t_1, t_2)$ where $T_1 \leq t_1 \leq t_2 \leq T_2$.*

Proof (i) Since $G \models q$ then in all models of G the distance between o and o' is less than or equal to D throughout the interval $[T_1, T_2]$. Then for any $d \geq D$; in all models of G the distance between o and o' is less than or equal to d throughout the interval $[T_1, T_2]$. Thus $G \models near(o, o', d, T_1, T_2)$.

(ii) Since $G \models q$ then in all models of G the distance between o and o' is less than or equal to D throughout the interval $[T_1, T_2]$. Then for any $[t_1, t_2] \in [T_1, T_2]$; in all models of G the distance between o and o' is less than or equal to D throughout the interval $[t_1, t_2]$. Thus $G \models near(o, o', D, t_1, t_2)$. $\square$

Theorem 15 *Let G be consistent, simple go theory and $q = near(o, o', d, t_1, t_2)$ be a near atom. Suppose $\Psi_{q,G}$ is a constraint set containing the following constraints:*

- $T_1 \leq t_1 \leq t_2 \leq T_2$
- $D \leq d$
- $\forall t \in [T_1, T_2], \Delta_G(o, o', t) \leq D$.

$\Psi_{q,G}$ is a solution to q w.r.t. G iff there is a substitution that satisfies $\Psi_{q,G}$.

Proof The theorem has two parts.

If part: If $\Psi_{q,G}$ is a solution to q w.r.t. G then there is a substitution that satisfies $\Psi_{q,G}$: This follows from the definition of a solution for a non-ground query.

Only if part: If there is a substitution that satisfies $\Psi_{q,G}$ then $\Psi_{q,G}$ is a solution to q w.r.t. G. This basically can be shown by the fact that the constraints in $\Psi_{q,G}$ are the necessary and sufficient conditions for satisfaction of a near atom. Let ϕ be a substitution that satisfy $\Psi_{q,G}$ such that $(t_1, v_1), (t_2, v_2)$ and (d, v_3) are in ϕ. Since ϕ satisfies $\Psi_{q,G}$, the constraints in $\Psi_{q,G}$ ensures that during the time interval $[v_1, v_2]$ the maximum distance between o and o' is less than or equal to v_3. It follows from the definition of satisfaction of a near atom that $G \models \phi(g)$. This is true for any ϕ that satisfies $\Psi_{q,G}$ hence, $\Psi_{q,G}$ is a solution to q w.r.t. G. $\Box$

Motion Closed World Assumption

Lemma 28 *A go theory G is coherent iff for every object o there is a spatially continuous plan π_o for G^o.*

Proof The proof trivially follows from Theorem 2 which requires for consistency the existence of a plan and Definition 45 requires for coherence the spatial continuity. $\Box$

Lemma 29 *Let G be a coherent go theory, o be an object and π be a spatially continuous plan for G^o. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements of π. Let $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$ be an atom. All instances of π that are coherent satisfy a iff there is an entry-exit (i, j) of $LS(\gamma_1) \ldots LS(\gamma_n)$ for $Vol(a)$ such that*

$$T^+(G^o, \pi, \gamma_i, P_i) \leq t_2 \text{ and } t_1 \leq T^-(G^o, \pi, \gamma_j, Q_j).$$

where $[P_k, Q_k] = LS(\gamma_k) \cap Vol(a)$,

Proof The proof of this lemma is very similar to proof of Lemma 12. It is a generalization of the case investigated in Lemma 12 for multiple movements. We omit the details to prevent repetition. $\Box$

Theorem 17 *Suppose G is a coherent* go*-theory and $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$ is a ground atom. Then: a is entailed by G via MCWA iff for every plan of G^o, the algorithm* $\mathsf{CheckCoherentIn}(G, \pi, a)$ *returns "true".*

Proof The proof of this theorem is very similar to proof of Theorem 8. It is a generalization of the case investigated in Theorem 8 for multiple movements. We omit the details to prevent repetition. $\Box$

Lemma 30 *Let G be a coherent go theory, o be an object and π be a spatially continuous plan for G^o. Let $\gamma_1 \sqsubseteq \gamma_2 \cdots \sqsubseteq \gamma_n$ be the movements of π. Let $a = \mathsf{in}(o, q_1, q_2, t_1, t_2)$ be an atom. All instances of π that are coherent satisfy $\neg a$ iff all of the following hold:*

- $T^+(G^o, \pi, \gamma_1, loc_1(\gamma_1)) \leq t_1$
- $T^-(G^o, \pi, \gamma_n, loc_2(\gamma_n)) \geq t_2$
- $\forall$ *entry-exit* (i, j) *of* $LS(\gamma_1) \ldots LS(\gamma_n)$ *for* $Vol(a)$,

  $T^-(G^o, \pi, \gamma_i, P_i) > t_2$ *or* $T^+(G^o, \pi, \gamma_j, Q_j) < t_1$

  where $[P_k, Q_k] = LS(\gamma_k) \cap Vol(a)$.

Proof The proof of this lemma follows from the definition of $\neg$in and Lemma 29. $\Box$

Theorem 18 *Suppose G is a coherent* go*-theory and $L = \neg\mathsf{in}(o, q_1, q_2, t_1, t_2)$ is a ground literal. Then L is entailed by G via* **MCWA** *iff for every plan of G^o, the algorithm* CheckCoherentNotIn(G, π, L) *(see Algorithm 8.17) returns "true".*

Proof The proof of this theorem is very similar to proof of Theorem 8. It is a generalization of the case investigated in Theorem 8 for multiple movements. We omit the details to prevent repetition. $\Box$

Deconfliction

Lemma 31 *Let $\langle G, \sqsubseteq \rangle$ be a coherent ordered-go-theory and C be a set of integrity constraints. If there is a deconfliction of $\langle G, \sqsubseteq \rangle$ w.r.t. C then for every integrity constraint $c \in C$ there is a candidate w.r.t. $\langle G, \sqsubseteq \rangle$.*

Proof Suppose there is a deconfliction, but there is a goal c with $Extent(c) = [t_1, t_2]$ which has no candidate. This could be because there is no spatial candidate or there is a spatial candidate which is not a candidate. In the first case, all go atoms have paths that are outside of $Vol(c)$. Since being a coherent model of $\langle G, \sqsubseteq \rangle$ requires satisfying atoms in $\langle G, \sqsubseteq \rangle$ and waiting at the end points of the atoms and it poses no additional constraint on where the vehicle will be at other times (i.e. before the first or after the last atom), we can easily construct a model of $\langle G, \sqsubseteq \rangle$ that stays out of $Vol(c)$ at all times. Any restriction of $\langle G, \sqsubseteq \rangle$ will have the same property so there can not be any restriction, that would entail c. Consider the second case where for all spatial candidates c, $Extent(c) \cap [t_1^-(g_i), t_2^+(g_j)] = \emptyset$. Then in all models of $\langle G, \sqsubseteq \rangle$, the vehicle will pass through $Vol(c)$ either before t_1 or after t_2. Thus there will always be a model that will stay out of $Vol(c)$ throughout $[t_1, t_2]$. This is also true in any restriction of $\langle G, \sqsubseteq \rangle$ so there can not be a deconfliction. Similar reasoning can be used for nogo regions. $\square$

Theorem 19 *Let $\langle G, \sqsubseteq \rangle$ be a coherent ordered-go-theory and o be an object and C be a set of integrity constraints. Suppose $g_1 \sqsubseteq^o g_2 \cdots \sqsubseteq^o g_n$ are the atoms of G^o. There is a deconfliction of $\langle G, \sqsubseteq \rangle$ w.r.t. $C[o]$ iff there is a $G'[o] = \bigcup_{i=1}^{n} go(o, loc_1(g_i), loc_2(g_i), t_{i1}, t_{i1}, t_{i2}, t_{i2}, v_i, v_i)$ such that $\langle G'[o], \sqsubseteq \rangle$ is a deconfliction of $\langle G^o, \sqsubseteq^o \rangle$ w.r.t. $C[o]$*

Proof Let Π^* be a deconfliction of $\langle G, \sqsubseteq \rangle$ w.r.t. $C[o]$ and $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models \Pi^*$. Then by Definition 4 and definition of a restriction, there are time intervals $[t_{11}, t_{12}] \ldots [t_{n1}, t_{n2}]$ such that $\mathcal{I}$ satisfies g_i over $[t_{i1}, t_{i2}]$. Let $\langle G'[o], \sqsubseteq^o \rangle$ be an ordered-go-theory such that $G'[o] = \bigcup_{i=1}^{n} go(o, loc_1(g_i), loc_2(g_i), t_{i1}, t_{i1}, t_{i2}, t_{i2}, v_i, v_i)$ and $v_i = dist(g_i)/(t_{i2} - t_{i1})$. Clearly $\langle G'[o], \sqsubseteq^o \rangle$ is a deconfliction of $\langle G^o, \sqsubseteq^o \rangle$ w.r.t. $C[o]$. The other direction of the statement trivially follows from definitions. $\square$

Lemma 32 *Let $\langle G, \sqsubseteq \rangle$ be a coherent ordered-go-theory and o be an object. Suppose $g_1 \sqsubseteq^o g_2 \cdots \sqsubseteq^o g_n$ are the atoms of G^o. Let ϕ be a solution of constraints $\mathcal{D}$ associated* *with* $\mathsf{DECON}(\langle G, \sqsubseteq \rangle, C, o)$. *Suppose* $G_o = \bigcup_{i=1}^{n} go(o, loc_1(g_i), loc_2(g_i), t_{i1}, t_{i1}, t_{i2}, t_{i2},$ $V_i, V_i)$ *is a set of go atoms where*

- $V_i = 1/\phi(v_i)$
- $t_{i1} = \phi(S_i)$
- $t_{i2} = t_{i1} + \phi(v_i) * dist(g_i))$.

Then $\langle G_o, \sqsubseteq^o \rangle$ *is a deconfliction for* $\langle G^o, \sqsubseteq^o \rangle$ *w.r.t* $C[o]$.

Proof Since the constraint set $\mathcal{D}$, includes all the constraints in $\mathcal{L}(G, o, \sqsubseteq^o)$, it is clear that $\Pi' = \langle G_o, \sqsubseteq^o \rangle$ satisfies the first condition of being a deconfliction. That is: all atoms in G_o is a restriction of another atom in G^o. Furthermore in every model $\mathcal{I}$ of Π', $\mathcal{I}$ satisfies g_i over the time interval $[t_{i1}, t_{i2}]$. Next we need to show that the candidate constraints added into $\mathcal{D}$, ensures that $\Pi' \models C[o]$. Let c be a goal constraint such that $Extent(c) = [t_1, t_2]$ and $\langle P, g_i, Q, g_j \rangle$ be the candidate selected in the algorithm. Then for any model $\mathcal{I}$ of Π', $\mathcal{I}(o, T_p) = P$ and $\mathcal{I}(o, T_Q) = Q$ where $T_P = t_{i1} + \mathsf{dist}(P, loc_1(g_i))/V_i$ and $T_Q = t_{j1} + \mathsf{dist}(Q, loc_1(g_j))/V_j$. The constraints in D enforces that $T_P \leq t_2$ and $T_Q \geq t_1$ and candidate definition ensures that that any point between P and Q will be in the $Vol(c)$ thus $\Pi' \models c$. Using a similar reasoning we can show that every nogo region in $C[o]$ is also entailed by $\langle G_o, \sqsubseteq^o \rangle$. $\Box$

Theorem 20 $\langle G, \sqsubseteq \rangle$ *has a deconfliction w.r.t.* C *iff for every vehicle* o *such that* $G^o \neq \emptyset$, DECON*(*$G, \sqsubseteq, C, o, \emptyset$*) returns "YES" for some nondeterministic trace. .*

Proof As shown in Lemma 32, when $\mathsf{DECON}(G, \sqsubseteq, C, o)$ returns YES the set of constraints $\mathcal{D}$ has a solution. Let $g_1 \sqsubseteq^o g_2 \cdots \sqsubseteq^o g_n$ be the atoms in G^o. It follows from

Theorem 19 that if a deconfliction for $\langle G^o, \sqsubseteq^o \ \rangle$ exists, then there is a constant speed, constant time restriction of G^o. Let $\Pi' = \langle G'[o], \sqsubseteq^o \ \rangle$ be a deconfliction such that, for all $g \in G'[o]$, $t_1^-(g) = t_1^+(g)$ and $t_2^-(g) = t_2^+(g)$ and $v^-(g) = v^+(g)$. Let $g'_1 \sqsubseteq^o g'_2 \cdots \sqsubseteq^o g'_n$ be the atoms in $G'[o]$. Because of these restrictions, every $\mathcal{I}$ that models Π', satisfies every $g'_i \in G'[o]$ over $[t_1^-(g_i), t_2^+(g_i)]$. Suppose c is a goal and $Extent(c) = [t_1, t_2]$. Since $\Pi' \models c$ and every model of Π' satisfy the atoms in the same intervals; there is a time point $t \in [t_1, t_2]$ such that for all $\mathcal{I} \models \Pi'$, $\mathcal{I}(o, t) = X \ and \ X \in Vol(c)$. Furthermore there is a $g'_x \in G'[o]$ such that $X \in LS(g'_x)$ and $t \in [t_1^-(g'_x), t_2^+(g'_x)]$. Let $\theta = \langle P, g_i, Q, g_j \rangle$ be the candidate such that $g_i \sqsubseteq g_x \sqsubseteq g_j$. Such a candidate exists because g'_x is a restriction of g_x and at least the point $X \in LS(g_x)$ is in $Vol(c)$. Also since $t \in [t_1^-(g'_x), t_2^+(g'_x)]$, $Extent(c) \cap [t_1^-(g_x), t_2^+(g_x)] \neq \emptyset$. Note that for any model $\mathcal{I}$ of Π', $\mathcal{I}(o, T_p) = P$ and $\mathcal{I}(o, T_Q) = Q$ where $T_P = t_1^-(g_i) + \mathsf{dist}(P, loc_1(g_i))/v^-(g_i)$ and $T_Q = t_1^-(g_j) + \mathsf{dist}(Q, loc_1(g_j))/v^-(g_j)$. Given this and the order on the atoms; we can conclude that $T_P \leq t_2$ and $T_Q \geq t_1$. It is easy to see that for every $g \in G^o$ when the variables S_g is $t_1^-(g')$ and v_g is $1/v^-(g')$ the constraints in $\mathcal{L}(G^o, o, \sqsubseteq^o)$ are satisfied. Furthermore when $X_P = T_P$ and $X_Q = T_Q$ the constraints in $\mathcal{C}(\theta, G, \sqsubseteq, c)$ are also satisfied. It is trivial to generalize this reasoning for other unsafe integrity constraints. Thus when Π' exists we can construct a constraint set $\mathcal{D}$ which has a solution. $\Box$