Applications of a Logic of Knowledge to Motion Planning under Uncertainty

Ronen I. Brafman
Department of Computer Science
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada
Tel: (604) 822-0594
Email: brafman@cs.ubc.ca
and
Jean-Claude Latombe
Robotics Laboratory
Department of Computer Science
Stanford University, Stanford, CA 94305, USA.
Tel: (415) 723-0350
Email: latombe@cs.stanford.edu.
and
Yoram Moses Department of Applied Mathematics and Computer Science
The Weizmann Institute of Science
Rehovot, 76100 Israel
Email: yoram@cs.weizmann.ac.il
and
Yoav Shoham Robotics Laboratory
Department of Computer Science
Stanford University, Stanford, CA 94305, USA.
Tel: (415) 723-3432
Email: shoham@cs.stanford.edu

Inspired by the success of the distributed computing community in applying logics of knowledge and time to reasoning about distributed protocols, we aim for a similarly powerful and high-level abstraction when reasoning about control problems involving uncertainty. This paper concentrates on robot motion planning with uncertainty in both control and sensing, a problem that has already been well studied within the robotics community. First, a new and natural problem in this domain is defined: Does there exists a sound and complete termination condition for a motion, given initial and goal locations? If yes, how to construct it? Then we define a high-level language, a logic of time and knowledge, which we use to reason about termination conditions and to state general conditions for the existence of sound and complete termination conditions in a broad domain. Finally, we show that sound termination conditions that are optimal in a precise sense provide a natural example of knowledge-based programs with multiple implementations.

Categories and Subject Descriptors: L.2.9 [**Artificial Intelligence**]: Robotics—*Motion Planning*; L.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Epistemic Logic*

General Terms: Design,Theory

Additional Key Words and Phrases: motion planning under uncertainty, logics of knowledge and time, analysis, knowledge representation

1. INTRODUCTION

Much research carried out in computer science in general, and AI in particular, is concerned with the development of powerful abstractions and their application to problems of interest. In the context of this article, of particular note is the use of logics of knowledge in distributed computing [Halpern and Moses 1990; Fagin et al. 1995]. The essential insight behind that line of research is that a formal notion of "knowing", initially developed in philosophy [Hintikka 1962] and later imported to AI [Moore 1985], can be coherently and usefully applied to reasoning about (and also designing) distributed protocols. The reasons for the success of this approach include:

Intuitiveness: Knowledge logics allow statements such as "Processor A doesn't know that processor B is faulty", which people use naturally to reason about distributed protocols.

Groundedness: The formal notion of knowledge is anything but vague; it is defined precisely in terms of the underlying protocol.

Abstraction and Generality: In principle, the notion of knowledge is dispensable. However, the analysis in terms of knowledge homes in on the essential notion, the knowledge available to the various processors at different points in time, and allows one to abstract away from the details of how the particular physical protocol implements that knowledge. This knowledge-level abstraction makes it possible to analyze and design protocols before their physical implementation is specified; in fact, the same knowledge-level protocol could be implemented differently, without affecting the high-level analysis.

However, while logics of knowledge have been widely used in AI (e.g., in human-computer interaction models, distributed planning, and nonmonotonic logics), they still have rarely been applied in correspondence to some specific concrete system. Besides distributed computing systems mentioned above, two other applications come to mind: Levesque's knowledge-level analysis of databases [Levesque 1984] and Rosenschein and Kaelbling's Situated Automata [Rosenschein 1985; Kaelbling and Rosenschein 1990]. We believe that planning and control in the presence of uncertainty is another wide arena in which such application can be particularly beneficial. In this article we take one step towards exploring this arena, concentrating on robot motion planning.

Robot motion planning under uncertainty is a well researched area [Lozano-Pérez et al. 1984; Canny 1989; Donald 1989; Erdmann 1986; Latombe 1991; Latombe et al. 1991]. Uncertainty may arise from several sources, including imperfect information on the locations of objects, sloppy robot control, and noisy sensing. A typical question asked in robotics is: "Given a goal region in which the robot must end up and bounds on both the slop in control and the noise in sensing, what are the regions from which the robot can start and be guaranteed to reach the goal and eventually recognize that it is in the goal?" These initials regions are called the *preimages* of the goal. To get a multi-step motion plan, this question is repeated in a backward-chaining fashion [Lozano-Pérez et al. 1984]. In this paper we assume motion commands of the form 'Go in direction D until the termination condition T is satisfied', and we investigate the following, seemingly more basic question: "Given an initial and a goal regions, a motion direction D, and bounds on the slop in control and noise in sensing, does there exist a *sound* and *complete* termination condition T?" By 'sound', we mean that T will only stop the motion in the goal; by 'complete', we mean that T will eventually stop the motion. We will also be interested in optimal sound termination conditions; such a condition evaluates to `true` no later than any other sound termination condition. Since one can naturally analyze a robot situation in terms like "the robot knows that it is in the goal, since it knows that the current sensory reading could only have been obtained if it were either at the goal or beyond the wall, and it knows that its motion plan could not possibly have led it behind the wall", investigating the above question using a logic of knowledge seems particularly well suited.

The following simple path planning problem in $\mathbb{R}$ is intended to give a better feel of the question addressed in this paper. We will also use it later to illustrate various definitions and results, along with other examples.

Example 1 Our robot is a point moving forward along the positive reals, starting at 0. It moves at unknown finite velocity until the termination condition is satisfied, at which point it stops. The goal is the interval $[2, 4]$. This domain is described in Figure 1. A sensor measures the robot's position with an uncertainty of 1, i.e., if the robot is at q, the sensor reading can be any value in $[q-1, q+1]$. At each time t, this sensor supplies a reading r, but the stream of sensed positions over time, i.e., the function $r(t)$, may not be continuous.

Clearly '$r > 1$' is a complete termination condition, but not a sound one. Similarly, '$r = 3$' is a sound termination condition, but not a complete one since the reading '3' may never occur due to discontinuities of $r(t)$. For instance, it is possible that the readings are accurate until the robot reaches 2.5 and then become consistently off by $+1$, i.e., jump from 2.5 to 3.5 and grow. Somewhat surprisingly, there exists a termination condition that is both sound and complete. We will derive it later from our theorems, but the reader may want to try to find it now. ▮

In [Mason 1984], Mason proposed the following synthesis of a most powerful termination condition:

- Prior to any motion, compute the set S of all possible robot trajectories consistent with the possible initial locations of the robot and the bound on control slop.
- During motion, at every time t, remove from S all trajectories that cannot produce the current reading given the bound on sensing noise; if all robot positions at time t along the trajectories remaining in S are in the goal, return `true`, else return `false`.

However, in virtually all practical cases this synthesis does not yield an effective termination condition, since the set S usually contains an infinite number of trajectories. Other researchers [Erdmann 1986; Latombe et al. 1991; Lazanas 1994] have proposed concrete algorithms for constructing termination conditions, but these algorithms either apply only to specific cases or do not yield the most powerful conditions. Hence, they may fail to generate a sound and complete termination condition, even if one exists. Our work lies between these two extremes. We offer general logic-based results on sound and complete termination conditions. In general, these results do not directly yield implementations of termination conditions, but we argue that they provide a more useful characterization of such conditions than [Mason 1984]. In particular, they allow us to state a necessary and sufficient condition for the existence of a sound and complete termination condition. Moreover, as we will see, in many cases they also help in the construction of termination conditions with desirable properties.

Another advantage of our tools is that they allow us to cover a wider class of motion planning problems. Mason's termination condition is based in an essential way on the assumption that the robot has complete knowledge of the history of its sensor readings. Implementing this termination condition in practice would require a robot to have a memory of unbounded size. In contrast, we are able to characterize optimal termination conditions for robots that have no memory. One application of this ability to characterize optimal termination conditions in different domains is demonstrated in Theorem 4, where we prove that memory is not essential in certain domains. That is, we show that when a sound and complete termination condition that relies on memory of past sensor values exists, then a sound and complete termination condition that relies only on current sensor value exists as well. Moreover, our results apply to goals that are non-geometric. For example, in a multi-robot system, a reasonable goal may be to reach some (geometric) goal region, while making sure that some other robot knows that we are in that region so that collisions are avoided. Hence, the goal here combines a geometric sub-goal with a non-geometric subgoal, and our results apply in this case as well.

This paper is intended for two audiences, robotics researchers interested in new tools for reasoning about and representing motion planning problems as well as new results in this domain, and researchers working on formal modeling and representation tools interested in new application domains. Consequently, we have tried to accompany the introduction and use of the formal tools with sufficient intuitive motivation and examples to help the former audience understand the essence of these tools and their applications, while providing sufficient explanations and examples of the motion planning problem to help the latter audience understand the application domain.

The paper is organized as follows. Section 2 defines the problem of finding sound and complete termination conditions for a classical, but particular class of motion planning settings. It states geometric conditions for the existence of such termination conditions and provides effective constructions of these termination conditions when they exist. Section 3 introduces the syntax and semantics of a logic of knowledge and time that will be used to study termination conditions in a wider class of settings. Section 4 performs such an analysis and offers a characterization of optimal sound termination conditions, as well as a study of optimal sound and complete conditions. Section 5 ends the body of this paper with a short discussion of knowledge-based programs. There, we show that optimal termination conditions are the first natural example of knowledge-based programs that have multiple implementations [Halpern and Fagin 1989; Fagin et al. 1995], and we rephrase the problem of generating such conditions as a problem of generating the optimal implementation of a knowledge-based program. Finally, Section 6 provides a discussion wrapping up our investigation.

Throughout this paper, proofs of the results are omitted from the main text; they appear in the Appendix.

2. TERMINATION CONDITIONS IN MOTION PLANNING

In this section we describe a rather specific class of motion planning problems with uncertainty and we formally define the problem of the existence of sound and complete termination conditions. Our description is based on the standard model used in the motion planning literature [Lozano-Pérez et al. 1984].

2.1 Class of problems

Figure 2 illustrates our description. Starting from any position within a given initial region $I \subset \mathbb{R}^n$ (a singleton in the figure), the robot is a point[1] commanded to move in a given direction D. The goal is a compact subset G of $\mathbb{R}^n$. Uncertainty in control and sensing are modeled as follows:

—At each time during a motion, the actual direction of motion of the robot may deviate from D by up to $\delta < \pi/2$, the *control uncertainty*. This deviation is not necessarily constant. In addition, the velocity of the robot must exceed some $\epsilon > 0$ but must be lower than some maximal velocity, v_B.

—The sensor measuring the robot's position returns a value that differs from the actual position by a distance of up to ρ, the *sensing uncertainty*. The sensor operates continuously, so that a sensed position is associated with each point in time. However, these sensed positions are not necessarily a continuous function of time, allowing us to model robots that do not continuously evaluate their sensor readings. For example, such robots can be modeled in this framework by attributing to them sensor readings that correspond to a step function.

Notice that given a single initial position s_i, the set of positions reachable from s_i given these bounds on the allowable motions is described by a cone (see Figure 2).

We use the term *motion planning instance*, denoted by ξ, to refer to the set of motions specified by δ, ρ and I. The *termination condition*, T, is a (total) Boolean function on the set of possible sensor readings. The first time it evaluates to `true` the robot stops.

Returning to Example 1, we see that the motion is in $\mathbb{R}$, the motion direction is $D = +$ (i.e. towards $+\infty$), the goal is $G = [2, 4]$, there is no control uncertainty (although, as we mentioned, the velocity is unknown), and the sensing uncertainty is $\rho = 1$.

We use annotated trajectories to describe the movement of the robot. Formally, an *annotated trajectory* is a pair $\tau = (Q, V)$, where Q is a continuous and piecewise differentiable function from $[0, +\infty)$ to $\mathbb{R}^n$, and V is a function from $[0, +\infty)$ to $\mathbb{R}^n$. The function Q describes the robot's position at any instant of time, while V describes the sensor readings.

---
[1] Any robot can be represented as a point in its configuration space [Lozano-Pérez 1983].

Definition 1. The annotated trajectory $\tau = (Q, V)$ is *consistent* with a given motion planning instance $\xi = (\delta, \rho, I)$ and a motion direction D if the following properties are satisfied:

—Q is continuous and piecewise differentiable;

—Q(0)$\in$ I;

—$|V(t) - Q(t)| \leq \rho$ for all $t \geq 0$;

—$|dQ(t) - D| \leq \delta$ for all $t \geq 0$, where $dQ(t)$ is the direction of the tangent to Q at t;[2]

—$\infty > v_B > |\dot{Q}(t)| > \epsilon > 0$.

Thus, an annotated trajectory is consistent with ξ if the robot's motion is described by a piecewise differentiable path commencing somewhere in I whose direction is consistent with the control uncertainty δ and whose velocity is bounded from above and below, and if the position readings are in line with the sensing uncertainty parameter ρ.

Definition 2. Given a motion planning instance $\xi = (\delta, \rho, I)$, a goal region G, and a motion direction D, a termination condition T is *sound* if, for every annotated trajectory $\tau = (Q, V)$ consistent with ξ and D, we have: for every $t \geq 0$, if $T(V(t))$ holds then there exists $t' \leq t$ such that $T(V(t'))$ holds and $Q(t') \in G$. T is *complete* if, for every annotated trajectory consistent with ξ and D, there exists $t \geq 0$ such that $T(V(t))$ holds.

This second definition deserves a comment. It says that a termination condition is sound if the first time it becomes `true` then the robot is in the goal. However, it is possible that no time at which T holds will be first. For example, this is the case if T is "the sensed position is greater than 1," the robot moves in $\mathbb{R}$ starting at 0 toward increasing abscissae, and sensing is perfect. T is nevertheless sound for the goal $[1, 2]$ because it satisfies the requirement that for every time t in which T is `true`, there is a time $t' \leq t$ in which both T is `true` and the robot is in the goal. In particular, because G is compact (and hence closed), if T is sound then $Q(\hat{t}) \in G$, where $\hat{t} = inf\{t \mid T(V(t))\}$.

Later on, we will also consider the notion of optimality. A sound termination condition is *optimal* if for each possible trajectory of the robot, it evaluates to `true` no later than any other sound termination condition. Thus, if there exists a sound and complete termination condition, the optimal sound termination condition must also be complete.

The model of [Lozano-Pérez et al. 1984], which we have adopted here, views the robot as moving continuously and evaluating its termination condition instantaneously. It also assumes that the robot can stop at once. While these assumptions may not always seem appropriate, they can be withdrawn by adjusting the value of ρ. For example, the distance a robot moves between the time it decides to stop and the time it actually stops can be added to ρ. Discrete evaluation of the termination condition can be taken into account in a similar way.

[2]Formally, this requirement applies only to the points t at which $Q(t)$ is differentiable. We make no requirement of this type on the finitely many points where $Q(t)$ is not differentiable.

Given this standard model, we consider the following two questions in this paper:

> *Given a motion planning instance ξ, a goal region G, and a motion direction D:*
> - *Does there exist a sound and complete termination condition?*
> - *What is the optimal sound termination condition?*

In the rest of this section we provide a partial answer to the above questions using geometric methods. It is a constructive answer albeit somewhat restricted. In Sections 3 through 5 we will introduce and use more powerful tools for reasoning about uncertainty, which will eventually allow us to characterize the class of optimal sound termination conditions.

2.2 The naive termination condition

From now on we consider a fixed given motion planning instance ξ, goal region G, and motion direction D.

Let the *forward projection* of a region R, denoted by $\mathcal{F}(R)$, be the set of all positions that can be reached from R by a path consistent with ξ and D, assuming $T \Leftrightarrow$ `false`. Hence, all positions q the robot may ever be at must satisfy $q \in \mathcal{F}(I)$.

Define $\Delta Q(r)$ to be the set of actual robot's positions at which a position reading of r is possible, independently of how the position was reached. $\Delta Q(r)$ is a disk of radius ρ centered at r. When the reading is r the robot's position, q, must be such that $q \in \Delta Q(r) \cap \mathcal{F}(I)$. In the context of ξ and D, the set $\Delta Q(r) \cap \mathcal{F}(I)$ depends only on r.

Definition 3. Given a goal region G, the condition $\Delta Q(r) \cap \mathcal{F}(I) \subset G$ is called the *naive termination condition.* It is denoted by $T_N(r)$.

Hence, the naive termination condition returns `true` given a reading r if all points reachable from I that are within a distance of ρ from r are within the goal region. Clearly, T_N is a sound termination condition.

In Example 1 the naive termination condition is '$r = 3$'. The following is a more complex example:

Example 2 The example is shown in Figure 3. The naive termination condition $T_N(r)$ is depicted as the region bounded by a dashed line in Figure 3; it is defined by 'the position reading r falls within this region'. Hence, we use the same figure to depict two spaces: actual positions and position readings. The full lines bound regions within the space of actual robot positions, while the dashed lines mark regions within the space of position readings.

Figure 3 shows two position readings r_1 and r_2, which are respectively outside and inside the termination condition. The actual positions where r_1 is possible are contained in $\Delta Q(r_1)$. Some of them (the shaded area) are not in the goal. Thus, a reading of r_1 may be obtained from outside the goal. Some of the positions where we can obtain r_2 are also outside the goal, but these positions are outside $\mathcal{F}(I)$; thus, they cannot be reached from I. ∎

The naive termination condition is not necessarily complete; for instance, as mentioned before, '$r = 3$' is not complete in the setting of Example 1. Similarly, in Example 2, if the distance between the points A and B (see Figure 3) was less than 2ρ, one could construct consistent annotated trajectories for which the naive termination condition would never evaluate to `true`.

2.3 Existence of a sound and complete termination condition

Using geometric arguments we now derive a number of conditions for the existence of a sound and complete termination condition in *domains containing no obstacles.* We still assume a fixed motion planning instance ξ, goal region G, and direction of motion D.

Definition 4. A *motion cutset* is a set of positions at least one of which must be traversed by every consistent trajectory. A *separating cutset* is a motion cutset **cs** such that $\mathcal{F}(I) \setminus \mathbf{cs}$ consists of two disjoint connected components, one of which includes the set I of initial positions.

Figure 4 shows an example of separating and non-separating cutsets.

Given a reading r, let $q(r)$ be the position whose value is r. We use this notation to emphasize the fact that the space of readings and the space of positions are distinct, although both are copies of $\mathbb{R}^n$. Correspondingly, given a termination condition T we define:

$$q(T) = \{q(r) \ \mid \ T(r)\}$$

THEOREM 1. *If $q(T_N)$ contains a separating cutset, then there exists a sound and complete termination condition.*

The constructive proof of this theorem (given in the Appendix) relies on the fact that we can disregard positions consistent with a sensor reading if we know that reaching them requires the robot to pass through earlier positions that satisfy the termination condition. This can be illustrated using Example 1.

Example 1 (continued) Since the space is $\mathbb{R}$, the naive termination condition '$r = 3$' is a separating cutset. The construction given in the proof of Theorem 1 yields the sound and complete termination condition '$r \in [3,5]$'. Although a reading in $[3,5]$ can be obtained with the robot outside the goal, it can be obtained for the *first* time only when the robot is in the goal. Indeed, any reading obtained at position 4 must satisfy the termination condition; therefore the robot cannot pass position 4 without its termination condition becoming `true` in the process. Moreover, the termination condition cannot evaluate to `true` before position 2. ∎

More generally, but still intuitively, the construction in the proof of Theorem 1 consists of the following: First "push" the separating cutset by a distance of ρ (in Example 1, where $\rho = 1$, this leads to pushing the separating cutset $\{3\}$ to $\{4\}$). This new set is a separating cutset as well, and the definition of T_N implies that it must be in the goal (and indeed, we see that $\{4\} \subset [2,4]$). Next, make this new set into a "trap" by including in the termination condition any reading that is possible when the robot is in this set. In Example 1, where we want to "trap" the robot no later than position 4, we use the termination condition "Stop if $r \in [3,5]$," which is guaranteed to evaluate to `true` when the robot is in position 4. Thus, if the robot enters the new cutset, it will get a reading that satisfies the termination condition; hence, this termination condition is complete. Because the new cutset has been obtained by pushing the old one by ρ, the termination condition cannot become `true` before the goal is achieved; hence, it is sound.

THEOREM 2. *If $q(T_N)$ is not a motion cutset and I is a singleton, then no sound and complete termination condition exists.*

Theorem 1 said that for a sound and complete termination condition to exist it is sufficient that $q(T_N)$ be a separating cutset. Theorem 2 tells us that $q(T_N)$ must be a motion cutset. When the goal is a convex polygon and the initial region is a single point, if $q(T_N)$ is a motion cutset then it is also a separating cutset. Thus:

COROLLARY 1. *If G is convex and I is a singleton, then a sound and complete termination condition exists if and only if $q(T_N)$ is a separating cutset.*

3. KNOWLEDGE-LEVEL FORMALIZATION

The previous geometric results are restricted to a rather specific class of motion planning problems (e.g., constant control uncertainty, position sensors exhibiting constant disk-like uncertainty). To overcome these limitations, we introduced (in [Brafman et al. 1993]) and later extended (in [Brafman et al. 1994]) a framework for reasoning about uncertainty in motion planning using a formal notion of knowledge. We present this framework below.

3.1 Runs and systems

We can view our endeavor as the investigation of a class of two-player protocols (see [Taylor et al. 1988]). Our players are the robot, which tries to follow the commanded motion direction, and the environment, which decides nondeterministically how the robot actually moves and what it senses, within given margins. This leads us to consider an instantaneous description of the state of affairs consisting of a state for the environment and a state for the robot.

Definition 5. Let $\mathcal{R}$ denote the set of possible (local) states of the robot and $\mathcal{E}$ denote the set of possible (local) states of the environment. A *global state* is a pair (e,r), where $e \in \mathcal{E}$ and $r \in \mathcal{R}$. The set of global states is denoted by $\mathcal{G}$.

In the domain of the previous section, the local state of the robot consists of its current position reading, while the local state of the environment consists of the robot's actual position. In this section the reader may still want to think about the local state of the environment as representing the actual position of the robot. The local state of the robot may then consist of its position reading, as before, or it may be different; e.g., it may contain information from other types of sensors or memory of past sensor readings. However, our model applies to more general control problems, where one attempts to recognize that a certain condition holds. For example, a controller monitoring a chemical process may need to recognize that the temperature is within a certain range before adding a new compound. The local state of the controller may then consist of a temperature reading, while the state of the environment may correspond to the actual temperature of the process.

The behavior of the robot and the environment over time is captured by *runs* associating a global state with every time.

Definition 6. A *run* R is a function from $[0, +\infty)$ to $\mathcal{G}$. We identify a *system* with a set of runs.

Hence, given a run R and a time $t \geq 0$, $\mathsf{R}(t)$ is a global state. It represents the state of the system at time t of the run R. We let $\mathsf{R}_a(t)$ and $\mathsf{R}_e(t)$ denote the robot's local state and the environment's local state, respectively, in the global state $\mathsf{R}(t)$. (Since we use r to denote sensor readings, both here and later on we use "a", for "agent", to refer to the robot.) The above definition of runs differs from the definition in [Fagin et al. 1995] in that time is continuous rather than discrete.

In general, we will focus on a subset of possible runs, those runs that describe the behaviors we would like to model. For example, in the domain of Section 2, the given motion planning instance ξ, goal region G, and motion direction D determine a subset of runs that correspond the annotated trajectories consistent with ξ and D. We denote this system by $S(\xi, D)$.

As in Section 2, we still require the robot's position to be a continuous and piecewise differentiable function of time since this is a property one would expect from actual trajectories. Termination conditions are total Boolean functions on the space of local states of the robot. The definitions of soundness and completeness are straightforward generalizations of our previous definitions.

Definition 7. Let S be a system and G be a subset of states of the environment. A termination condition T is *sound* for S and G if for every run $\mathsf{R} \in S$ and time t, if T evaluates to `true` in $\mathsf{R}_a(t)$, then there exists a time $t' \leq t$ such that T evaluates to `true` in $\mathsf{R}_a(t')$ and $\mathsf{R}_e(t') \in G$. A termination condition T is *complete* for S if for every run $\mathsf{R} \in S$ there exists a time t, such that T evaluates to `true` in $\mathsf{R}_a(t')$.

3.2 A language and its semantics

We now present a language for reasoning about termination conditions. This language contains temporal and epistemic modal operators and has intuitive semantics.

We assume some propositional language $\mathcal{L}$, including a primitive proposition g whose intuitive reading is 'the robot is in the goal'. We also assume that for every termination condition T of interest, $\mathcal{L}$ contains a primitive proposition T. Given the set $\mathcal{G}$ of global states of a system S, an *interpretation* π for $\mathcal{L}$ in $\mathcal{G}$ is a function assigning a truth value to every primitive proposition of $\mathcal{L}$ at every state in $\mathcal{G}$. We call the pair $\mathcal{I} = (S, \pi)$ an *interpreted system*.

We define the satisfaction of a formula φ in $\mathcal{L}$ with respect to $\mathcal{I}$ at a global state $s \in \mathcal{G}$ (written $\mathcal{I}, s \models \varphi$) as follows:

—if φ is a primitive proposition, then $\mathcal{I}, s \models \varphi$ if and only if $\pi(s, \varphi) =$ `true`;

—if $\varphi = \neg\psi$, then $\mathcal{I}, s \models \varphi$ if and only if $\mathcal{I}, s \not\models \psi$;

—if $\varphi = \psi_1 \wedge \psi_2$, then $\mathcal{I}, s \models \varphi$ if and only if $\mathcal{I}, s \models \psi_1$ and $\mathcal{I}, s \models \psi_2$.

We use the notation $\mathcal{I} \models \varphi$ (read φ is *valid* in $\mathcal{I}$) when φ is satisfied by all global states that occur in (runs of) $\mathcal{I}$, and $\models \varphi$ (read φ is *valid*) when $\mathcal{I} \models \varphi$ for all interpreted systems $\mathcal{I}$.

We assume that π is 'natural', e.g., the proposition g is satisfied exactly at those states in which the robot is in the goal, and the proposition T is satisfied exactly when the condition T holds. Since we assume that π is fixed, we will refrain from mentioning π explicitly in every system we consider. Hence, a run R of a system S will also be a run of the interpreted system $\mathcal{I} = (S, \pi)$.

Motions occur over time. Therefore, we add temporal operators allowing us to reason about what has happened in the past or what will happen in the future of a given state. We use the so-called *branching-time* temporal operators, which allow us to quantify over the possible futures or pasts of a given global state. Temporal logics of this type have been extensively investigated by various researchers. For details see, e.g., [Emerson and Halpern 1985], who introduced this syntax, and [Halpern and Vardi 1986], whose semantics we use. Branching-time temporal operators combine two types of quantification: over the set of possible futures, and over the states of a particular future branch. Thus, the $\forall$ or $\exists$ part of the operator tells us whether we are universally or existentially quantifying over the set of possible futures, while the $\Box$ or $\Diamond$ part tells us whether we are universally ($\Box$) or existentially ($\Diamond$) quantifying over the states of each particular future. For example, $\forall_{\Box}$ talks about all states ($\Box$) within all possible futures ($\forall$), while $\exists_{\Box}$ talks about all states ($\Box$) within some future branch ($\exists$), and $\exists_{\Diamond}$ talks about some future state ($\Diamond$) in some future branch ($\exists$). In order to refer to the set of past branches instead of future branches, we put a $-$ sign inside the $\Box$ or $\Diamond$, as in $\forall_{\boxminus}$, $\exists_{\boxminus}$, and $\exists_{\diamondminus}$. Finally, in the above quantifiers, the present state is considered to be part of both the past and the future. When we wish to speak about strict past or strict future, we add a bar over $\Box$ and $\Diamond$, as in $\forall_{\overline{\boxminus}}$ and $\forall_{\overline{\Box}}$. All together there are 16 temporal operators.

Here is the precise definition of a subset of the set of operators which we use in this paper:

—**There exists a future in which φ always holds:** We define $\mathcal{I}, s \models \exists_{\Box}\varphi$ to hold if there exists a run R of $\mathcal{I}$ and a time t such that $\mathsf{R}(t) = s$ and for all $t' \geq t$ we have $\mathcal{I}, \mathsf{R}(t') \models \varphi$.

—**For all pasts, φ sometimes holds:** We define $\mathcal{I}, s \models \forall_{\diamondminus}\varphi$ to hold if for all runs R of $\mathcal{I}$ and times t, if $\mathsf{R}(t) = s$, then there exists a time $t' \leq t$ for which $\mathcal{I}, \mathsf{R}(t') \models \varphi$.

—**There exists a strict past in which φ always holds:** We define $\mathcal{I}, s \models \exists_{\overline{\boxminus}}\varphi$ to hold if there exists a run R of $\mathcal{I}$ and a time t such that $\mathsf{R}(t) = s$ and for all $t' < t$ we have that $\mathcal{I}, \mathsf{R}(t') \models \varphi$.

Example 3 We illustrate two of the operators in Figure 5. In this figure we see three possible choices of pasts for the state s and three possible choices of futures for that state. Thus there are nine possible runs, all commencing in I, that contain the state s. Each such run contains nine consecutive states, some of which are marked as satisfying one of two propositions, p and q. We see that any choice of past for state s contains some state in which p is satisfied. Hence

$$\mathcal{I}, s \models \forall_{\diamondminus} p$$

Notice that this formula can be read also as "p must have held sometime in the past."

In addition, we see one possible future for s in which q is always satisfied. Hence,

$$\mathcal{I}, s \models \exists_{\Box} q$$

This formula can be read also as "it is possible that q will always hold from now on." ∎

To deal with uncertainty we formally define the notion of knowledge as follows:

Definition 8. Two global states s and s' are *indistinguishable* to the robot, denoted by $s \sim_a s'$, if the local state of the robot is identical in s and s'.

Recalling our first example of motion in $\mathbb{R}$, there we have that $(3, 2.1) \sim (2.4, 2.1)$. These states are indistinguishable to the robot because it obtains identical readings in both of these global states.

Note that $\sim_a$ is an equivalence relation.

Definition 9. Let $\varphi \in \mathcal{L}$. The robot *knows* φ at a state s in the system $\mathcal{I}$, written

$$\mathcal{I}, s \models K_a\varphi$$

if $\mathcal{I}, s' \models \varphi$ holds for every state s' in $\mathcal{I}$ such that $s \sim_a s'$.

Hence, the robot knows φ at $s = (e, r)$ if the local state r can only occur when φ holds.

Example 4 Consider Example 1 again. There, we have that

$$\mathcal{I}, (3.5, 3) \models K_a g$$

That is, at the global state $(3.5, 3)$, the robot knows that it is in the goal because the set of states $s = (q, r)$ that are indistinguishable from $(3.5, 3)$ is $\{(q, 3) \mid |q-3| \leq 1\}$, or equivalently, $\{(q, 3) \mid 2 \leq q \leq 4\}$. In all these states, the robot's position is in the goal. ∎

From now on we will assume that the language $\mathcal{L}$ is closed under the 16 temporal operators and the epistemic operator K_a. All definitions remain unchanged. We remark that our definition of knowledge is the standard one used for knowledge in distributed systems (see [Fagin et al. 1995]) and is well-known to satisfy the modal system S5. Note that once we fix a system $\mathcal{I}$ of interest, the satisfaction in state s of a formula of the form $K_a\varphi$ depends only on the local state of the robot. Thus, $K_a\varphi$ can be used to define a termination condition (although not necessarily an easily computable one).

Finally, note that knowledge, as we use it here, is an ascribed notion providing a tool for external analysis of termination conditions, as would be appropriate for the designer of a robot to perform. We do not view this as something the robot will use for its own reasoning.

4. OPTIMAL TERMINATION CONDITIONS

We now wish to apply the framework defined in Section 3 to the analysis of termination conditions. The framework allows us to study motion termination at an abstract level by investigating properties of general systems, which can correspond to quite different assumptions on our robot and its domain. Therefore, our results will hopefully be applicable to many concrete contexts of interest.

4.1 The motion planning context

Roughly speaking, we would like to cast the question of designing optimal termination conditions in terms of the following question:

> *Having fixed all relevant parameters other than the termination condition, when does a sound and complete termination condition exist? If a "best" termination condition exists, what form will it have?*

Because our ability to speak about arbitrary sets of runs gives great flexibility in modeling different contexts, this generalizes the question we asked in Section 2. In general, to incorporate various assumptions on the setting investigated, we consider eight main parameters that can be varied in the specification of the system:

(1) The workspace: The space in which the robot moves may contain various obstacles or have various geometric properties. It may contain non-geometrical dimensions, e.g., the temperature of a gas.

(2) The goal region G: It may be a connected or non-connected subset of the workspace.

(3) Knowledge about the initial position: This will be given in terms of a set of possible initial states. In a motion planning instance ξ this is captured by I.

(4) The specified direction: While we previously assumed a fixed motion direction D, we may in general imagine a more elaborate specification of direction which could depend, for example, on the robot's sensor readings and the position of the goal, or on the actual position (due, for example, to changes in the local orientation of the surface the robot is traversing).

(5) Control uncertainty: In a motion planning instance ξ this was captured by δ, the maximal possible deviation of the actual direction from the commanded motion direction D. More generally, the control uncertainty may change, possibly growing over time, or varying depending on the robot's current position.

(6) Sensing uncertainty: In a motion planning instance ξ, at any given position, the robot's possible position readings could be within a disk of radius ρ centered at this actual position. Now the possible readings may depend on time and on the robot's position. For example, the robot's position readings may be less accurate the farther it is from its initial state. Sensing is also not necessarily restricted to position, and may include other parameters such as forces or temperatures. Sensing memory may be employed as well.

(7) Progress guarantees: It is usually hard to ensure nontrivial properties for a robot that is not guaranteed to move. It is therefore customary to make certain assumptions regarding the robot's progress. For example, our definition of a consistent trajectory in Section 2 required that the robot's velocity be greater than some $\epsilon > 0$ as long as it has not explicitly halted. We call this a *progress guarantee.* A *weak progress guarantee*, which is sufficient in many applications, is that the robot moves an infinite distance over an infinite amount of time, unless it explicitly halts after a finite amount of time.

(8) Bounded velocity: The velocity of the robot is bounded from above by some value v_B.

We call a collection of assumptions describing the above parameters a *motion planning context*. Notice that varying any of these parameters (except perhaps the goal region) will change the set of possible trajectories. Varying the goal region will change the meaning of the soundness of a termination condition. The contexts discussed in Section 2 were ones determined by a motion planning instance $\xi = (\delta, \rho, I)$, a goal region G, a fixed direction D, and the progress guarantee that the robot velocity is always greater than some $\epsilon > 0$.

Every motion planning context κ determines a system $\mathcal{I}(\kappa)$ consisting of all runs consistent with the assumptions in κ when the robot never performs an explicit halting action (i.e., all termination conditions are set to `false`); we call this the *uninhibited* system. Once we fix a motion planning context, and hence, the agent's configuration space, its actuation, sensing, and memory capabilities, we also fix an uninhibited system. This system can then be used to study and compare the effect of applying different termination conditions. The only assumption that is inherent to this approach is that no aspect of any run, prior to stopping, is affected by the termination condition. In particular, we can use the language introduced earlier to characterize and reason about termination conditions: Having fixed an uninhibited system, we have also fixed the semantics of formulas in this language. In particular, the meaning of formulas containing the knowledge operator, whose semantics is system-based rather than run-based, is fixed as well.

Definition 10. Given an uninhibited system $\mathcal{I}$, a termination condition T is *as strong* as another termination condition T' if:

$$\mathcal{I} \models T' \to \forall_{\diamond} T.$$

Given a class $\mathcal{T}$ of termination conditions, T is *optimal* (with respect to $\mathcal{T}$) if it is as strong as any other $T' \in \mathcal{T}$.

In words, T is *as strong as* T' if whenever T' holds, it is the case that T must have held sometime in the past, or holds now. For example, w.r.t. the domain of Example 1, the termination condition $r \in [3, 6]$ is as strong as the termination condition $r \in [3, 5]$, but the termination condition $r = 2$ is not as strong as $r = 2.5$, nor is $r = 2.5$ as strong as $r = 2$. For each of these termination conditions, there are runs in which one is satisfied before the other.

One useful observation given is the following:

Lemma 1. *Let $\mathcal{I}$ be an uninhibited system and G be a goal.*

(a) There is an optimal sound termination condition.

(b) If a sound and complete termination condition exists, then the optimal sound termination condition is also complete.

(c) There exists an optimal sound termination condition T such that $\mathcal{I} \models T' \to T$ for any other sound termination condition T'.

Notice that Part (c) of this lemma tells us that we can assume that the optimal sound termination condition is always satisfied whenever any other sound termination condition is satisfied. However, according to our definitions, an optimal sound termination condition need not have this property, since we do not care what values the condition obtains after the first time that it is true. An optimal sound termination condition is required only to be true no later than any other sound termination conditions, for every possible trajectory.

4.2 Optimal sound and complete termination conditions

At this stage we have adequate tools to characterize soundness and completeness. Let $\mathcal{I}$ be an uninhibited system, I the initial region, G the goal region, g a proposition that is satisfied precisely when the robot is in G, and T a termination condition.

Lemma 2. *(a) T is sound if and only if $\mathcal{I} \models T \rightarrow \forall_{\diamondsuit}(T \wedge g)$.*

(b) T is complete if and only if for all global states s at which the robot is in I, we have: $\mathcal{I}, s \models \forall_{\diamond} T$.

This lemma shows us that the definitions of soundness and completeness can be captured within our language. For example, Part (a) tells us that T is sound if whenever T holds T and g must have held (together) in the past or the present (Recall that $\forall_{\diamondsuit}$ refers to both past and present.)

We now state a theorem that characterizes the optimal sound and complete termination conditions for a large class of domains:

Theorem 3. *If $\mathcal{I} \models (\neg g \wedge \forall_{\diamondsuit} g) \rightarrow \exists_{\square} \neg g$ and a sound and complete termination condition exists, then $K_a \forall_{\diamondsuit} g$ is the optimal sound and complete termination condition.*

In words, $\mathcal{I} \models (\neg g \wedge \forall_{\diamondsuit} g) \rightarrow \exists_{\square} \neg g$ means: For every global state s in every run R of $\mathcal{I}$, if the robot is not in the goal in s, but has been previously in the goal, then it is possible that the robot will never enter the goal again. On the other hand, $K_a \forall_{\diamondsuit} g$ means: The robot knows that it (either is or) has been in the goal. Recall that the satisfaction of any formula of the form $K\varphi$ depends only on the local state of the robot, and hence, represents a termination condition. As formally proven in the Appendix, $K_a \forall_{\diamondsuit} g$ eventually evaluates to `true` if there exists a sound and complete termination condition; hence, it is complete. In addition, provided that a sound and complete termination condition exists and $\mathcal{I} \models (\neg g \wedge \forall_{\diamondsuit} g) \rightarrow \exists_{\square} \neg g$, the first time $K_a \forall_{\diamondsuit} g$ evaluates to `true` the robot is necessarily in the goal; hence it is sound.

Note that Theorem 3 does not make specific assumptions about the control and sensing uncertainties, the shape of the set of initial positions or the goal region, or about the progress guarantees. In fact, the theorem applies even in contexts in which the various parameters are correlated in various ways. On the other hand, although it gives a constructive definition of an optimal candidate for a sound and complete termination condition, verifying that this termination condition holds at a particular state and that it is indeed sound and complete may be computationally expensive. However, for domains in which sensing uncertainty is constant disk-like and control uncertainty is constant, our experience indicates that this is a a simple task. We would expect it to remain so, as long as the sensing and control uncertainty have a simple geometric description. The use of knowledge has led to optimal solutions of an analogous flavor in fault-tolerant distributed systems [Dwork and Moses 1990; Moses and Tuttle 1988].

The following lemma considers motion planning contexts that generalize the motion planning instances of Section 2 by relaxing the restrictions on sensing uncertainty and progress guarantee, and shows that when the goal is convex, the condition used in Theorem 3 holds.

LEMMA 3. *Let κ be a motion planning context defined by parameters $\delta, \Delta V, I, G, D$ and the weak progress guarantee, where δ, I are defined as in a motion planning instance, G is a goal region, D is a direction of motion, and $\Delta V : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is an arbitrary function that maps a position q of the robot to the set of possible readings at q. If the goal region G is convex and there are no obstacles in the workspace, then*

$$\mathcal{I}(\kappa) \models (\neg g \wedge \forall_{\diamond} g) \rightarrow \exists_{\square} \neg g.$$

Example 1 (continued) The goal [2,4] obviously satisfies the condition $\mathcal{I}(\kappa) \models (\neg g \wedge \forall_{\diamond} g) \rightarrow \exists_{\square} \neg g$. (Indeed, in this case, not only is it possible never to enter the goal after exiting it, it is necessarily the case.) Clearly, in this example the conditions of Lemma 3 are satisfied.

The condition $\forall_{\diamond} g$ is satisfied by all positions $q \geq 2$. Therefore, $K_a \forall_{\diamond} g$ is satisfied by readings $r \geq 3$. Thus, the condition '$r \geq 3$' is our candidate for an optimal sound and complete termination condition. We have seen in Section 2.3 that the termination condition $r \in [3, 5]$ is sound and complete. Using the same line of reasoning, it is easy to see that the termination condition $r \geq 3$ is also sound and complete. ▮

Example 2 (continued) The domain described in Figure 2 satisfies the condition $\mathcal{I}(\kappa) \models (\neg g \wedge \forall_{\diamond} g) \rightarrow \exists_{\square} \neg g$ as well. Indeed, after exiting the goal, it is always possible for the robot to follow a trajectory that never re-enters the goal. Alternatively, we can use Lemma 3 to obtain this result. Theorem 3 yields the optimal sound and complete termination condition depicted in Figure 6 (hatched region). ▮

Example 5 This third example illustrates the need for the condition $\mathcal{I} \models (\neg g \wedge \forall_{\diamond} g) \rightarrow \exists_{\square} \neg g$. G is made of two disconnected regions G_1 and G_2 such that all uninhibited trajectories from I traverse both G_1 and G_2, and G_1 is entered first. Assume that $T \Leftrightarrow K_a \forall_{\diamond} g$ always halts the robot in G_1. Now consider the same context with the following difference: the position sensor has a time delay λ so that at every time t it returns the measurement of the robot's position at time $t - \lambda$. Let the upper bound on the velocity v_B be large enough and G_1 small enough so that, for some trajectories, when $K_a \forall_{\diamond} g$ evaluates to `true` for the first time, the robot is no longer in G_1, but not yet in G_2. However, let v_B be small enough and G_2 large enough so that for every trajectory there is a time when $K_a \forall_{\diamond} g$ evaluates to `true` and the robot is in G_2. In this context, which does not satisfy $\mathcal{I} \models (\neg g \wedge \forall_{\diamond} g) \rightarrow \exists_{\square} \neg g$ (although there exists a sound and complete termination condition), $K_a \forall_{\diamond} g$ is complete, but not sound. ▮

Example 6 Another example illustrating the need for this condition appears in Figure 7. There, G consists of two disconnected regions and $\mathcal{I} \not\models (\neg g \wedge \forall_{\diamond} g) \rightarrow$

$\exists_{\square}\neg g$. As we will see later, there is a sound and complete termination condition for this domain, but it is not $K_a \forall_{\diamond} g$. ∎

Example 7 Theorem 3 can be applied when the goal does not correspond directly to a geometric goal region. Consider the case of a robot $\mathtt{a}$ that moves in a domain containing other mobile robots. This robot is not only concerned with arriving at a (geometric) goal region G, but is also concerned with collision avoidance. For that reason, the robot's actual goal may be not only to arrive at G, but to do so while ensuring that some other robot $\mathtt{b}$ is aware of the fact that $\mathtt{a}$ is in G. This goal can be described by the formula $\hat{g} \equiv g \wedge K_{\mathtt{b}} g$.[3] (As before, we take g to be a proposition that is true exactly when the robot $\mathtt{a}$ is in the geometric region G.) Because $K_{\mathtt{b}} g \rightarrow g$ is a valid formula, we have that $\hat{g} \equiv K_{\mathtt{b}} g$. This formula may not correspond to a geometric condition (because it refers to the set of local states of robot $\mathtt{b}$, on which a natural metric need not exist, or which could be discrete). Nevertheless, we can still apply Theorem 3 in this case. Now, the condition we have to check first is whether $\mathcal{I} \models (\neg K_{\mathtt{b}} g \wedge \forall_{\diamond} K_{\mathtt{b}} g) \rightarrow \exists_{\square} \neg K_{\mathtt{b}} g$, and the candidate for an optimal termination condition is $K_{\mathtt{a}} \forall_{\diamond} K_{\mathtt{b}} g$. As in Theorem 3, additional computation is still required to obtain an actual termination condition from this characterization. Yet, we already have a non-trivial knowledge-level characterization to guide us. ∎

We have not established a precise correspondence between Theorems 1 and 3. However, it seems that if in Theorem 1 we construct our termination condition based on the earliest cutset within $q(T_N)$ (where the notion of earlier is defined much like the notion of optimality) we should obtain a termination condition whose power is similar to that of the termination condition given in Theorem 3.

The position we took so far was that almost all aspects or our problem are fixed. That is, the configuration space, the agent's sensors and actuators, its motion direction, and its memory were fixed. Our goal was to supply powerful termination conditions for this fixed agent. However, it is natural to vary other aspects of the robot description. In particular, we would like to see what role memory plays in the generation of sound and complete termination conditions. To this end, we now examine a robot whose configuration space, actuation and sensing capabilities, as well as its motion direction are fixed. However, now we allow ourselves to vary its termination condition *and* memory capabilities. Once we allow ourselves to vary the agent's memory capabilities, the description of its local state, and hence the description of global states, changes. Hence, for each memory capability given to this robot, we will have to use a different uninhibited system.

Additional memory of past readings can supply the robot with more knowledge, i.e., it can rule out more worlds as impossible. This can allow it to know various facts that it would not know based only on its current reading. However, we now

[3] A careful treatment of this example would require details that are beyond the scope of the presentation in this paper. First of all, we have not defined knowledge in the multi-agent case. However, the definition of $K_{\mathtt{a}}$ and $K_{\mathtt{b}}$ in the multi-agent case is standard (see [Fagin et al. 1995]), and coincides with the definition of knowledge in the single-agent case. Moreover, the goal of $K_{\mathtt{b}} g$ makes sense with respect to a particular system, which should either be fixed, or about which we should be able to make assumptions (e.g., that $\mathtt{b}$ sees any movement on the Northern side of a given obstacle).

show that in certain cases memory is not required to design sound and complete termination conditions. That is, in these contexts, if a sound and complete termination condition exists for the robot given that it maintains all of its past readings, then a sound and complete termination condition exists for that robot even if it maintains only its current position reading.

Let us say that a robot *has sensing history* if its local state at time t consists of the pair (t, h), where t is the current time and h is a function $h : [0, t] \to \mathbb{R}^n$ (or more generally, from $[0, t]$ to the space of sensor readings) assigning to each time $t' \in [0, t]$ the readings of the sensor at t'. (Notice that this involves an assumption that the robot has a clock.) We let $\mathcal{I}_h(\kappa)$ denote the uninhibited system in which the robot has sensing history in context κ.

THEOREM 4. *Let κ be a motion planning context defined by a motion planning instance $\xi = (\delta, \rho, I)$, where I is a singleton, in a space containing no obstacles, by a goal region G, by the motion direction D, and by a weak progress guarantee. Assume that (i) G's boundary and every possible path of the robot can be described by a finite number of algebraic curves; (ii) we have that $\mathcal{I}_h(\kappa) \models (\neg g \wedge \forall_{\diamond} g) \to \exists_{\square} \neg g$; and (iii) a sound and complete termination condition T for the robot in system $\mathcal{I}_h(\kappa)$ exists. Then there exists a sound and complete termination condition T' that depends only on the* current *sensor reading.*

In proving this theorem we use the following strategy: Theorem 3 provides an ideal candidate for a sound and complete termination condition for the robot without memory: $K_a \forall_{\diamond} g$ (evaluated w.r.t. the system in which the robot has no memory). We show that if this termination condition is not sound, then the termination condition $K_a \forall_{\diamond} g$ evaluated w.r.t. the system in which the robot maintains sensing history is not sound and complete either. This latter proof depends crucially on the fact that this domain is assumed to be free of obstacles. Since Theorem 3 tells us that $K_a \forall_{\diamond} g$ is the optimal sound and complete termination condition for that system when one exists, the fact that it is not sound refutes our initial assumption that a sound and complete termination condition based on complete sensing history exists.

Notice, that we could not follow this strategy if we were not able to characterize the *optimal* termination condition, nor could we use Mason's characterization of optimal termination conditions here because it is not constructive and does not apply to the case of no memory. Hence, we see this theorem as a good example of the utility of our approach.

4.3 Optimal sound termination conditions

Theorem 3 provides the optimal sound and complete termination condition *if one exists*; if this is not the case, the condition it provides may not even be sound. To see this, suppose we shrink the goal region in Example 1 to the set $[2, 3]$. $K_a \forall_{\diamond} g$ will still be satisfied in $[3, +\infty)$, yet it is clearly not a sound termination condition.

In the absence of a sound and complete termination condition, one may ask the following question: What is the optimal sound termination condition? If we give an answer to this question, we can then answer our original question by checking whether the optimal sound termination condition is complete. As stated in Lemma 1, if a sound and complete termination condition exist, then the optimal sound termination condition must be complete. Moreover, since the condition `false` is sound, the class of sound termination conditions is never empty, so that an optimal sound condition is always guaranteed to exist.

We now give a general characterization of the optimal sound termination condition. However, due to the use of fixpoints, this characterization does not have the same constructive flavor as the termination condition given by Theorem 3.[4] We will have more to say on this issue later. Given a set of states Y, we define Ψ_Y to be the formula that is true at the states of Y and false otherwise. We can now close our language under a new modal operator T, whose definition is as follows.

Definition 11. Given a system $\mathcal{I}$, for each $\varphi \in \mathcal{L}$, the formula $\mathsf{T}\varphi$ is defined to be Ψ_T for the largest set of states T such that $\mathcal{I} \models \Psi_T \Leftrightarrow K_a(\varphi \vee \forall_{\diamond}(\Psi_T \wedge \varphi))$.

In words, the above definition tells us that $\mathsf{T}\varphi$ is true at the states of the largest set T with the following property: $s \in T$ iff for every $s' \sim s$ (i.e. for every s' that the robot cannot distinguish from s) it is either the case that φ holds in s' or that on every run taking us from the initial state to s, there is a state that is in T in which φ holds. This definition captures the intuition that when the robot considers what positions are possible given its current readings, it should disregard positions that cannot be reached without passing through states in which the termination condition will be satisfied.

The theory of fixpoints tells us that a largest such set T exists for every formula φ if the map $x \mapsto K_a(\varphi \vee \forall_{\diamond}(x \wedge \varphi))$ is monotone, i.e., if $\mathcal{I} \models \alpha \rightarrow \beta$ then $\mathcal{I} \models K_a(\varphi \vee \forall_{\diamond}(\alpha \wedge \varphi)) \rightarrow K_a(\varphi \vee \forall_{\diamond}(\beta \wedge \varphi))$. This is indeed the case.

Let κ be a motion planning context, $\mathcal{I}(\kappa)$ the corresponding context, G the goal region, and g the proposition corresponding to the robot's being in G.

THEOREM 5. *The condition* $\mathsf{T}g$ *is an optimal sound termination condition.*

An immediate corollary of this result is:

COROLLARY 2. *A sound and complete termination condition exists if and only if* $\mathsf{T}g$ *is a complete termination condition.*

While the fixpoint definition may be useful for reasoning about optimal termination conditions, it does not indicate how to compute them. The following is a step in that direction:

THEOREM 6. *Let* $\mathcal{I}$ *be a system. Define* $\mathsf{T}_\sigma\varphi$ *as follows:*

—$\mathsf{T}_0\varphi \Leftrightarrow$ `true`*;*
—$\mathsf{T}_{\sigma+1}\varphi \Leftrightarrow K_a(\varphi \vee \forall_{\diamond}(\mathsf{T}_\sigma\varphi \wedge \varphi))$ *for successor ordinals;*
—$\mathsf{T}_\sigma\varphi \Leftrightarrow \bigwedge_{\tau<\sigma} \mathsf{T}_\tau\varphi$ *for limit ordinals.*

There exists an ordinal σ *for which* $\mathcal{I} \models \mathsf{T}\varphi \Leftrightarrow \mathsf{T}_\sigma\varphi$.

This theorem gives an iterative method for calculating T. We start with a very strong termination condition, `true`, which (unless $I \subset G$) is not sound. We progressively weaken this termination condition by substituting it into the formula

[4] While a technical introduction to the theory of fixpoints is beyond the scope of this paper, we believe that most of what follows does not require previous exposure to these tools. The interested reader is referred to [Tarski 1955; Halpern and Moses 1990] or [Fagin et al. 1995].

$K_a(\varphi \vee \forall_{\diamond}(x \wedge \varphi))$. It is a well-known result of the theory of fixpoints that (because this mapping is monotone) we will eventually reach a termination condition that is precisely $\mathsf{T}\varphi$.

Unfortunately, there is no useful bound on the number of iterations required for this process to reach this greatest fixpoint. However, we believe that for many reasonable domains, e.g., domains with polygonal goals and obstacles, only a small number of iterations will be needed. Support for this conjecture is suggested by noticing that $\mathsf{T}_1 g \Leftrightarrow K_a(g \vee \forall_{\diamond} g) \Leftrightarrow K_a \forall_{\diamond} g$, which is the termination condition given by Theorem 3. Thus, very roughly, Theorem 3 can now be understood as a claim regarding the rate of convergence of T_σ toward T. The condition used in Theorem 3, $\neg g \wedge \forall_{\diamond} g \rightarrow \exists_{\Box} g$, requires that once a trajectory has left the goal it will possibly never re-enter it. A more general condition would refer to a minimal number k of re-entries to the goal. We conjecture that the number of iterations required to obtain an optimal sound termination condition in this case is linear in k.

Example 8 In Figure 7 we see an example of a goal region consisting of two connected components. The above iterative procedure successively yields the following three termination conditions, displayed in Figure 8:

$$T_1 \stackrel{\text{def}}{=} K_a \forall_{\diamond} g,$$

$$T_2 \stackrel{\text{def}}{=} K_a(g \vee \forall_{\diamond}((K_a \forall_{\diamond} g) \wedge g)),$$

$$T_3 \stackrel{\text{def}}{=} K_a(g \vee \forall_{\diamond}(K_a(g \vee \forall_{\diamond}((K_a \forall_{\diamond} g) \wedge g)) \wedge g))$$

Neither T_1 nor T_2 is sound. In each case, Figure 8 displays a path for which readings exist such that the robot would stop at the last point of this path, which is outside the goal. However, T_3 is sound. ▮

5. RELATION TO KNOWLEDGE-BASED PROGRAMS

We now consider the problem of designing optimal sound and complete termination conditions from a slightly different point of view. This new view relates to the notion of a *knowledge-based program* and its implementations, as introduced in [Fagin et al. 1995]. For the purpose of the analysis in this section, we restrict attention to only systems and termination conditions such that for every run there is a first time when the termination condition is satisfied. For example, we obtain this property if we discretize time and a single local state of the robot is attained at each time instant. But the property can also be satisfied by systems where time is continuous.

Let $\mathcal{I}$ be an uninhibited system. A termination condition T induces a *different* system $\mathcal{I}[T]$ describing the runs of the robot in the same context. For each run R of $\mathcal{I}$, there is a run $\mathsf{R}[T]$ of $\mathcal{I}[T]$ that coincides with R up to the first time $\hat{t}$ at which T holds in R. (If no such point exists, then $\mathsf{R}[T] = \mathsf{R}$.) From that point on, the robot's position in $\mathsf{R}[T]$ remains fixed, as does its local state. $\mathcal{I}[T]$ is defined to be $\{\mathsf{R}[T] \mid \mathsf{R} \in \mathcal{I}\}$.

LEMMA 4. *(a) If T is sound w.r.t. $\mathcal{I}$ then $\mathcal{I}[T] \models T \rightarrow g$.*
(b) T is sound w.r.t. $\mathcal{I}$ if and only if $\mathcal{I}[T] \models T \rightarrow K_a g$.

(c) If T is an optimal sound termination condition, then $\mathcal{I}[T] \models T \Leftrightarrow K_a g$.

Item (b) resembles Theorem 3 in that it characterizes an optimal sound and complete termination condition T in terms of knowledge. In this case, however, the characterization is in terms of the knowledge the robot has in the system inhibited by T, instead of the uninhibited system. In the terminology of [Fagin et al. 1995], this lemma can be viewed as stating that an optimal sound termination condition is an implementation of the knowledge-based program Pg:

if $K_a g$ **then** halt.

This program is said to be *knowledge-based* because it contains a test—of whether $K_a g$ holds—that depends on the robot's knowledge. In comparison, a 'standard' program is one in which all tests are formulated directly in terms of variables and values that appear explicitly in the robot's local state. A standard program Pg^s will be said to *implement* Pg (with respect to an uninhibited system $\mathcal{I}$) if Pg^s has the form

if T holds **then** halt

where T is a condition that depends solely on the robot's local state, and $\mathcal{I}[T] \models T \Leftrightarrow K_a g$. In other words, if with respect to the system $\mathcal{I}[T]$ that represents the situation in which the robot follows Pg^s, the robot is indeed halting at the first instant in which it knows it is in the goal region. This is consistent with the intuitive reading of Pg.

We could hope to use Lemma 4 in order to obtain an alternative characterization of optimal sound and complete termination conditions. Indeed, we can show:

LEMMA 5. *Given G and κ, if the program $\mathsf{P_g}$ has a unique implementation in the context κ, then the condition defined by $K_a g$ in the system $\mathcal{I}_g$ representing $\mathsf{P_g}$ in the context κ is an optimal sound termination condition for $\mathcal{I}(\kappa)$ and G.*

Unfortunately, knowledge-based programs are not always guaranteed to have a unique (up to equivalence) implementation in terms of a standard program. There are examples of knowledge-based programs that can be implemented in inherently different ways in a given context. Nevertheless, most of these examples are somewhat unnatural, in some essential way making use of knowledge about the robot's future history to determine its current action. Fagin et al. [Fagin et al. 1995] provide a sufficient condition for the existence of a unique standard implementation of a knowledge-based program P in a given context κ. While the formal definition of this condition is somewhat technical, it applies to many applications of knowledge-based programs. Interestingly, even in fairly well-behaved motion planning contexts, the program $\mathsf{P_g}$ described above is not guaranteed to have a unique implementation.

Example 1 (continued) The following are two implementations of the knowledge-based program $\mathsf{P_g}$ in the context defined by this example:

(1) **if** $r = 3$ **then** halt.

(2) **if** $r \geq 3$ **then** halt.

Let us denote the condition '$r = 3$' by T_1 and '$r \geq 3$' by T_2. One can check that $\mathcal{I}[T_1] \models K_a(in[2,4]) \Leftrightarrow (r = 3)$ and $\mathcal{I}[T_2] \models K_a(in[2,4]) \Leftrightarrow (r \geq 3)$. In words, in the system $\mathcal{I}[T_1]$, the formula $K_a(in[2,4])$ coincides with the condition $r = 3$, while in $\mathcal{I}[T_2]$ it coincides with $r \geq 3$.

For example, if we examine the system $\mathcal{I}[T_2]$, we see that it does not include any run in which the robot's position is greater than 4. This implies that in any state in which the robot's reading is greater than or equal to 3, $K_a(in[2,4])$ holds. To see this, suppose that $r \geq 3$. In principle, r could be obtained from any position $q \in [r-1, r+1]$. However, since states in which the robot's position q is greater than 4 do not exist in this system, given a reading $r \geq 3$ the robot only considers possible positions in $[r-1, 4] \subseteq [2,4]$.

Intuitively, in the system $\mathcal{I}[T_2]$ the robot "knows" it is following the condition T_2. The reasoning behind the condition T_2 could be viewed as the robot knowing that "if the condition T_2 has been satisfied in the past, then the robot is in the goal". Thus, although in the uninhibited system a reading of, say, $r = 5$ would not ensure that the robot is in the goal region, in the system $\mathcal{I}[T_2]$ it does. ∎

The fact that the implementation of $\mathsf{P_g}$ in the above example is not unique does not depend on the robot's movement being continuous. A discrete analogue of this example can be shown to display the exact same property. In fact, inspired by the work reported here, a discrete analogue of this example is given in Chapter 7 of [Fagin et al. 1995].

In the discrete case, adding to the robot's local state a discrete "local clock", whose value is guaranteed to increase every time the global state changes, is enough to ensure that the test for $K_a g$ that appears in $\mathsf{P_g}$ will satisfy the sufficient conditions for a unique implementation given in [Fagin et al. 1995]. It is an interesting open question whether adding a local clock in the continuous case could guarantee a unique implementation of $\mathsf{P_g}$.

6. DISCUSSION

In this paper we considered a number of approaches to the question of the existence of sound and complete termination conditions in motion planning. As discussed in Section 5, the task of designing termination conditions can be thought of as the task of implementing an abstract knowledge-based program that tells the robot to stop once it knows that it is in the goal. However, the basic knowledge used to construct this termination condition can be different. For example, one may simply rely on the basic description of the sensor, i.e., the reading uncertainty, as is the case in the termination condition '$r = 3$' of Example 1 (where we only use the fact that the sensor may indicate a reading that is off by at most 1 from the true position). But there is additional information to be used. For example, the initial state and the motion direction are both taken into account by the naive termination condition, since we consider as impossible those positions that are outside the forward projection of the initial region. Finally, we can take into account the termination condition itself, considering a position to be impossible if every trajectory leading to it would cause the termination condition to become true prior to the arrival at this position. This consideration is manifested in the fixpoint nature of the operator T.

This paper is an example of the power and applicability of abstract logical concepts with concrete and intuitive semantics. The notions of knowledge and time describe important properties of the set of possible behaviors of a robot with imprecise sensors. They are based on a natural state-based description of the robot-environment system and allow us to concentrate on the particular issue of interest, that is, achieving a motion-related goal in an uncertain environment. The temporal-epistemic logic we introduced captures these notions of time and knowledge and provides a natural and powerful tool in the domain of motion planning with uncertainty, enabling us to characterize the optimal termination conditions in a large class of motion planning problems. Using it, we have been able to characterize optimal sound termination conditions in a class of domains wider than that to which Mason's description of optimal termination conditions applies [Mason 1984] and in a more constructive manner. We have taken advantage of both these properties in showing that in certain domains, memory is not required for sound and complete termination.

In order to practically use the abstraction provided by the high-level language used in this paper, we must bridge the gap between it and more concrete domain descriptions. We believe that an important task for future research will be to look for interesting temporal/epistemic properties of different sensors and domains (relating the knowledge level and the geometric level). Such properties can help us reason about more restricted classes of domains, hopefully obtaining stronger concrete results.

Current tools used in robotics to reason about uncertainty make very strong use of geometric analysis, from which considerable advantages can be obtained. Our logical approach is intended to supplement these tools when a high level analysis of issues pertaining to uncertainty is warranted. Moreover, while geometric reasoning is natural for single-agent motion planning problems with uncertainty, it does not seem an adequate tool for reasoning about an agent's uncertainty about the behavior of other agents. For example, in a multi-robot motion planning problem, a robot can be uncertain about its current position and the current position of the other robots, both of which can be given geometric description in terms of subsets of the configuration space. However, the robot will also be uncertain about the other robots' state of information. This uncertainty has important consequences for the robot's behavior because the state of information of the other robots determines their behavior. For example, it could determine their motion direction, and thus, which choice of direction by the robot will lead to (or be guaranteed to avoid) collision. Alternatively, the robot may actively seek to effect the other robots' state of information, as demonstrated in Example 7, in which case, the description of its goal is no longer purely geometric.

Knowledge, on the other hand, can be applied to many other natural problems, and it is especially suitable for the case of multiple agents, as its application in distributed systems demonstrates; its abstract nature enables uniform treatments of uncertainty of geometric and non-geometric nature.

Not surprisingly, the notion of knowledge is not completely new to robotics, and recent work by Erdmann [Erdmann 1994] uses a similar notion to reason about the sensing requirements of a task and suggests methods for designing such sensors. While Erdmann does not employ a formal language for reasoning about knowledge and time, the semantic notions employed in his work are closely related to the semantics of our notion of knowledge. Similarly, Donald [Donald 1994] discusses relationships among different sensors and sensor configurations, and we believe that knowledge plays an important role in these issues: equivalent sensors must supply us with identical knowledge. Initial steps in the exploration of these relations are taken in [Brafman and Shoham 1995].

Finally, sometimes we are willing to settle for less than knowledge. That is, we are willing to relax the soundness requirement and only demand that when we stop we are 'pretty sure' that we are in the goal. For this purpose, it seems likely that using some notion of belief, rather than knowledge, will be appropriate.

APPENDIX

A. PROOFS OF RESULTS IN SECTION 2

In Section 2 we made the following assumptions: both the motion planning instance ξ and the direction of motion D are given and fixed; the modulus of the robot's velocity is upper and lower bounded; the goal region G is a compact subset of $\mathbb{R}^n$ (hence, it is closed and bounded); and there are no obstacles.

Recall that for any termination condition T, $q(T)$ denotes the set of all positions coinciding with readings satisfying T.

Let σ be any $(n-1)$-dimensional surface patch. We denote by $S_\rho(\sigma)$ the volume swept out by an n-ball of radius ρ whose center spans σ. Whenever σ is continuous, the boundary of $S_\rho(\sigma)$ is a continuous surface homeomorphic to the boundary of an n-ball. E.g., in two dimensions it is a simple closed-loop curve (Jordan curve).

THEOREM 1. *If $q(T_N)$ contains a separating cutset, then there exists a sound and complete termination condition.*

PROOF. The proof is stated using terms appropriate for $\mathbb{R}^2$. In $\mathbb{R}^n$ one should substitute "$(n-1)$-dimensional surface" for "curve".

If $q(T_N)$ contains a separating cutset of $\mathcal{F}(I)$, then it necessarily contains a simple continuous curve segment (or a collection of such segments, which may happen only if I consists of several disconnected components) forming a separating cutset of $\mathcal{F}(I)$. Let us call σ one such a cutset. For simplification, we assume below that σ consists of a single curve segment, but the generalization to multiple segments is straightforward.

Let σ^+ denote the set of points in $\mathcal{F}(\sigma)$ (the forward projection of σ) whose distance to σ is ρ. We define the termination condition $\hat{T}$ as the set of readings possible from σ^+. We have: $\{q(r) \,|\, \hat{T}(r)\} = S_\rho(\sigma^+)$.

We show below that $\hat{T}$ is both complete and sound:

- *$\hat{T}$ is complete:* $S_\rho(\sigma)$ is a separating cutset of $\mathcal{F}(I)$. The lower portion of the intersection of its boundary $\partial S_\rho(\sigma)$ with $\mathcal{F}(I)$, i.e., $\partial S_\rho(\sigma) \cap \mathcal{F}(\sigma)$, is exactly σ^+. It is a simple continuous curve segment forming another separating cutset of $\mathcal{F}(I)$. Therefore, $\hat{T}$ is complete.
- *$\hat{T}$ is sound:* For every possible trajectory, a reading satisfying $\hat{T}$ is obtained no later than when σ^+ is reached. Therefore, using $\hat{T}$, the robot can never reach a point beyond σ^+. Now consider a reading r' such that $\hat{T}(r')$ holds. The robot's position must then be within distance ρ from $q(\hat{T})$, hence within 2ρ from σ^+, and it must also be before σ^+. Thus, it is within ρ from σ. By definition of T_N, every point that is no further away from a point in $q(T_N)$ than ρ is in G. Since $\sigma \subset q(T_N)$, the robot must be in G when $\hat{T}$ holds for the first time. Hence, $\hat{T}$ is sound. □

THEOREM 2. *If $q(T_N)$ is not a motion cutset and I is a singleton then no sound and complete termination condition exists.*

PROOF. Assume that a given termination condition T is sound and complete and that $q(T_N)$ is not a motion cutset. Our proof consists of showing that these two assumptions are incompatible. We will proceed by constructing a consistent annotated trajectory (Q_1, R_1) such that if t_1 is the infimum of times at which T holds along this trajectory, then $Q_1(t_1)$, the robot's position at t_1, is outside the goal. Because the robot's trajectory is a continuous function of time, this implies that the robot's position at t_1 is also the limit of the robot's positions where T holds. Since G is closed, this construction implies that T is not sound, for otherwise, $Q_1(t_1) \in G$.

Under the assumption that $q(T_N)$ is not a motion cutset, there exists a trajectory Q_0 that never passes through $q(T_N)$. Consider the particular consistent annotated trajectory (Q_0, R_0) in which all the readings are accurate, meaning that $R_0 = Q_0$. Since T is also assumed complete, there should exist a time at which T holds on (Q_0, R_0). Let $t_0 = \inf\{t : T(R_0(t))\}$ and $q_0 = Q_0(t_0)$. We have $q_0 \notin q(T_N)$. This does not mean that $q_0 \notin G$, but it does imply that $\mathcal{F}(I)$ contains a position $q_1 \notin G$ where the reading $r = q_0$ is possible (hence, $|q_1 - q_0| \leq \rho$).

We now construct the trajectory (Q_1, R_1) introduced above, such that $Q_1(t_1) = q_1$ and $t_1 = \inf\{t : T(R_1(t))\}$. The path of Q_1 beyond q_1 is parallel to the path of Q_0 beyond q_0. The path of Q_1 between I and q_1 is defined backward from q_1. It is parallel to Q_0's path between q_0 and I, until it hits the boundary of $\mathcal{F}(I)$; then it follows a straight-line segment toward I along $\mathcal{F}(I)$'s boundary. More formally, let us introduce the trajectory $\hat{Q}(t) = Q_0(t) + c(t)(q_1 - q_0)$, where:

$$c(t) = \max\{\alpha \mid \alpha \leq 1 \text{ and } Q_0(t) + \alpha(q_1 - q_0) \in \mathcal{F}(I)\}.$$

Note that $\hat{Q}(t_0) = q_1$ and that for all t, $|\hat{Q}(t) - Q_0(t)| \leq |q_0 - q_1| \leq \rho$. Note also that, if there exists $t' \in [0, t_0]$ such that $\hat{Q}(t')$ is in $\mathcal{F}(I)$'s boundary, then, for any $t \leq t'$, $\hat{Q}(t)$ is also in $\mathcal{F}(I)$'s boundary. Therefore, $\hat{Q}(t)$ is continuous and piecewise differentiable. If $\hat{Q}$ does not satisfy the velocity requirements, we can always reparameterize it using a differentiable function h, such that $\hat{Q}(h(t))$ satisfies these requirements. We then substitute $\hat{Q}(h(t))$ for $\hat{Q}(t)$.

To define R_1, let us first consider the case where $q_1 \notin \mathcal{F}(q_0)$; then $\hat{Q}(0) = I$ and we define $Q_1 = \hat{Q}$. Let $t_1 = h(t_0)$. For $t < t_1$ we set $R_1(t) = R_0(h(t)) = Q_0(h(t))$. For $t \geq t_1$ we set $R_1(t) = Q_0(t - t_1 + t_0)$. Clearly, $|Q_1(t) - R_1(t)| \leq |q_0 - q_1| \leq \rho$ for all t; hence, (Q_1, R_1) is a consistent trajectory. By construction, $\inf\{t : T(R_1(t))\} = t_1$ and $q_1 = Q_1(t_1) \notin G$.

In the case where $q_1 \in \mathcal{F}(q_0)$, then $\hat{Q}(0) \neq I$; but $\hat{Q}(0) \in \mathcal{F}(I)$. $Q_1(t)$ is obtained by concatenating two trajectories: the first part consists of a straight-line segment connecting I to $\hat{Q}(0)$ at an admissible velocity; the second part is defined by $Q_1(t) = \hat{Q}(t - t_s)$ for $t \geq t_s$, where t_s is the time when Q_1 reaches $\hat{Q}(0)$. We set $R_1(t)$ to be equal to $Q_0(0)$ throughout the added initial line segment. □

COROLLARY 1. *If G is convex and I is a singleton, then a sound and complete termination condition exists if and only if $q(T_N)$ is a separating cutset.*

PROOF. Following Theorems 1 and 2 it is sufficient to show that if $q(T_N)$ is a motion cutset then it contains a separating cutset σ.

We construct σ as follows. Consider an arbitrary ray R erected from I and contained in $\mathcal{F}(I)$. Let A and B be the first intersection points of R with G and $q(T_N)$, respectively. The existence of both points is guaranteed by the fact that $q(T_N)$ is a motion cutset. When R spans the cone $\mathcal{F}(I)$, A and B span two $(n-1)$-dimensional surface patches. We let γ denote the surface patch spanned by A (it is the portion of G's boundary in $\mathcal{F}(I)$ visible from I). The surface patch spanned by B is σ.

To establish that σ is a separating cutset, it suffices to show that it is continuous. For that purpose, consider any ray R specified as above. By definition of $q(T_N)$, the point where R intersects σ is exactly at distance ρ from γ. Hence, σ is the lower portion of the boundary of $S_\rho(\gamma)$ in $\mathcal{F}(I)$. The convexity of G implies the continuity of γ, which in turn implies that of σ. $\square$

B. PROOFS OF RESULTS IN SECTION 4

LEMMA 1. *Let $\mathcal{I}$ be an uninhibited system and G be a goal region. With respect to $\mathcal{I}$ and G,*

(a) there is an optimal sound termination condition;

(b) if a sound and complete termination condition exists, then the optimal sound termination condition is also complete;

(c) there exists an optimal sound termination condition T such that $\mathcal{I} \models T' \to T$ for every other sound termination condition T'.

PROOF. (a) First, we observe that the class of sound termination conditions is closed under disjunctions. Indeed, let $\{T_j : j \in J\}$ be a (possibly infinite) set of sound termination conditions, the termination condition $T = \bigvee_{j\in J} T_j$ is sound. In order to see this, consider an arbitrary run R. If $\bigvee_{j\in J} T_j$ is satisfied in $\mathsf{R}(t)$, then it must be the case that $T_{j_0}(t)$ is satisfied for some $j_0 \in J$. Because T_{j_0} is sound, either g holds, or there exists $t' < t$ such that $g \wedge T_{j_0}$ holds in $\mathsf{R}(t')$, in which case $g \wedge \bigvee_{j\in J} T_j$ also holds in $\mathsf{R}(t')$.

It follows that the termination condition $\hat{T} = \bigvee_{T' \text{ is sound}} T'$ is sound. To show that $\hat{T}$ is optimal, we must prove that $\mathcal{I} \models T \to \forall_{\diamondminus}\hat{T}$ for every sound termination condition T. However, if T is sound, then by definition of $\hat{T}$, we have that $\mathcal{I} \models T \to \hat{T}$, which implies $\mathcal{I} \models T \to \forall_{\diamondminus}\hat{T}$.

(b) If T is sound and complete, then in particular it is sound. Hence, the optimal sound termination condition $\hat{T}$ returns `true` no later than T. Thus, $\hat{T}$ is also complete.

(c) The termination condition $\hat{T}$ constructed in (a) has this property. $\square$

LEMMA 2. *(a) T is sound if and only if $\mathcal{I} \models T \to \forall_{\diamondminus}(T \wedge g)$.*

(b) T is complete if and only if for all global states s at which the robot is in I, we have: $\mathcal{I}, s \models \forall_{\diamond} T$.

PROOF. (a) If T is sound then, by definition, $\mathcal{I} \models T \to (g \vee \forall_{\diamondminus}(T \wedge g))$. However, $T \to (g \vee \forall_{\diamondminus}(T \wedge g))$ is logically equivalent to $T \to ((T \wedge g) \vee \forall_{\diamondminus}(T \wedge g))$, and $(T \wedge g) \vee \forall_{\diamondminus}(T \wedge g)$ is logically equivalent to $\forall_{\diamondminus}(T \wedge g)$. Thus, $\mathcal{I} \models T \to \forall_{\diamondminus}(T \wedge g)$.

(b) By definition, T is complete if in every run it eventually returns `true`. This is precisely the semantics of the condition $\mathcal{I}, s \models \forall_{\diamond} T$ for every $s \in I$. $\square$

THEOREM 3. *If $\mathcal{I} \models (\neg g \wedge \forall_{\diamondminus} g) \to \exists_{\Box}\neg g$ and a sound and complete termination condition exists, then $K_a\forall_{\diamondminus} g$ is the optimal sound and complete termination condition.*

PROOF. Let T be an optimal sound and complete termination condition.

- $K_a\forall_{\diamondminus} g$ is complete: Since T is sound, we have that $\mathcal{I} \models T \to \forall_{\diamondminus}(T \wedge g)$ and in particular $\mathcal{I} \models T \to \forall_{\diamondminus} g$. As T depends only on the robot's local state, the definition of K_a implies that $\mathcal{I} \models T \rightarrow K_a T$. Hence, $\mathcal{I} \models T \rightarrow K_a(\forall_{\diamondminus} g)$. Thus, T's completeness implies that $K_a \forall_{\diamondminus} g$ is complete.

• $K_a \forall_{\diamondminus} g$ is sound: Assume that $\mathcal{I}, s \models K_a(\forall_{\diamondminus} g) \wedge \neg g$. We know that $\mathcal{I}, s \models \forall_{\diamondminus} g$ (since $\models K_a \varphi \rightarrow \varphi$). Using $\mathcal{I} \models (\neg g \wedge \forall_{\diamondminus} g) \rightarrow \exists_{\Box} \neg g$, we obtain $\mathcal{I}, s \models \exists_{\Box} \neg g$. But T is both sound and complete, so it must be the case that $\mathcal{I}, s \models \forall_{\overline{\diamondminus}} T$. (Otherwise, because $\mathcal{I}, s \models \exists_{\boxminus} \neg g \wedge \exists_{\Box} \neg g$, there is a consistent trajectory in which $T \wedge g$ is never satisfied).

Thus, whenever $K_a(\forall_{\diamondminus} g) \wedge \neg g$ is satisfied in a run, there is an earlier state in that run in which T is satisfied. Because T is sound, this implies that whenever $K_a(\forall_{\diamondminus} g) \wedge \neg g$ is satisfied in a run, there is an earlier state in which $T \wedge g$ is satisfied. However, since we have just shown that $\mathcal{I} \models T \rightarrow K_a(\forall_{\diamondminus} g)$, we can conclude that whenever $K_a(\forall_{\diamondminus} g) \wedge \neg g$ is satisfied in a run, there is an earlier state in which $K_a(\forall_{\diamondminus} g) \wedge g$ is satisfied. Hence $K_a(\forall_{\diamondminus} g)$ is a sound termination condition. $\Box$

LEMMA 3. *Let κ be a motion planning context defined by parameters $\delta, \Delta V, I, G, D$ and the weak progress guarantee, where δ, I are defined as in a motion planning instance, G is a goal region, D is a direction of motion, and $\Delta V : \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$ is an arbitrary function that maps every possible position q of the robot to the set of possible readings at q. If the goal region G is convex and there are no obstacles in the workspace, then*

$$\mathcal{I}(\kappa) \models (\neg g \wedge \forall_{\diamondminus} g) \rightarrow \exists_{\Box} \neg g.$$

PROOF. Assume that for some state s we have $\mathcal{I}(\kappa), s \models \neg g \wedge \forall_{\diamondminus} g \wedge \forall_{\Diamond} g$, and let q_s be the robot's position in s. Assuming fixed control uncertainty δ, no obstacles, and a weak progress guarantee, we can find two positions in G, such that the line connecting them passes through q_s. By convexity of G this means that $q_s \in G$, and hence $\mathcal{I}(\kappa), s \models g$.

An explicit construction is as follows: let q be some position reachable from I that is in G and is before q_s (i.e., q_s is in the forward projection of q). The existence of such a position is guaranteed by the fact that $\mathcal{I}(\kappa), s \models \forall_{\diamondminus} g$. Let R be a run in which the trajectory of the robot moves from q to q_s directly (i.e. via a straight segment) and follows the same directions from q_s onwards. Such a trajectory is possible because δ (the control uncertainty) is constant and there are no obstacles in the domain. Since $\mathcal{I}(\kappa), s \models \forall_{\Diamond} g$, we know that somewhere along this run following q_s, there is a position $q_1 \in G$. We then use convexity to conclude that all points on this trajectory between q and q_1 are in G. In particular, $q_s \in G$ and hence $\mathcal{I}(\kappa), s \models g$. This contradicts our initial assumption that $\mathcal{I}(\kappa), s \models \neg g \wedge \forall_{\diamondminus} g \wedge \forall_{\Diamond} g$. $\Box$

THEOREM 4. *Let κ be a motion planning context defined by a motion planning instance $\xi = (\delta, \rho, I)$, where I is a singleton, in a space containing no obstacles, by a goal region G, by the motion direction D, and by a weak progress guarantee. Assume that (i) G's boundary and every possible path of the robot can be described by a finite number of algebraic curves; (ii) we have that $\mathcal{I}_h(\kappa) \models (\neg g \wedge \forall_{\diamondminus} g) \rightarrow \exists_{\Box} \neg g$; and (iii) a sound and complete termination condition T for the robot in system $\mathcal{I}_h(\kappa)$ exists. Then there exists a sound and complete termination condition T' that depends only on the* current *sensor reading.*

Remark: In this theorem we are comparing two uninhibited systems in which the local states of the robot contain different information. In one system, $\mathcal{I}(\kappa)$, the local state contains only the current position reading; in the other, $\mathcal{I}_h(\kappa)$, it contains full sensing history, which we denote by h. The sensing history h is a function of time returning the robot's position reading at each point. We let (q, r) denote states in the system $\mathcal{I}(\kappa)$ that only record the current reading, and we let (q, h) denote states in the system $\mathcal{I}_h(\kappa)$ where sensing history is maintained.

Notice that there is a one-to-one correspondence between the runs of these two systems: each run of the robot without memory defines precisely one run of the robot with memory, and vice versa. However, a state in $\mathcal{I}_h(\kappa)$ contain some information about the past which is not contained the corresponding state of $\mathcal{I}(\kappa)$. Indeed, consider two corresponding runs, one from each system, yielding the same trajectory and the same readings at all times. Let s be some state in the run of $\mathcal{I}(\kappa)$ and s_h be the corresponding state in the run of $\mathcal{I}_h(\kappa)$. Each of these states can occur in a number of runs. s_h can occur in runs in which the robot's trajectory is different, but in which the sensing history is identical; while s can occur in runs in which both the trajectory and sensing history differ, except for the current reading. Hence, there are usually fewer possible pasts for a state in $\mathcal{I}_h(\kappa)$ than for a state in $\mathcal{I}(\kappa)$. Thus, it may be the case that $\mathcal{I}_h(\kappa), s_h \models \neg g \wedge \forall_{\Diamond} g$, while $\mathcal{I}(\kappa), s \not\models \neg g \wedge \forall_{\Diamond} g$, and therefore, the condition $\mathcal{I}_h(\kappa) \models (\neg g \wedge \forall_{\Diamond} g) \rightarrow \exists_{\Box} \neg g$ is stronger than the condition $\mathcal{I}(\kappa) \models (\neg g \wedge \forall_{\Diamond} g) \rightarrow \exists_{\Box} \neg g$. The theorem requires the stronger condition to hold.

We first prove the following claim:

CLAIM 1. *Under the conditions of the theorem, assume that* $\mathcal{I}(\kappa), (q, r) \models \exists_{\boxdot} \neg K_a \forall_{\Diamond} g$. *Then there exists a sensing history h that the robot can have at position q, such that in h the reading obtained at q is r and* $\mathcal{I}_h(\kappa), (q, h) \models \exists_{\boxdot} \neg K_a \forall_{\Diamond} g$.

PROOF. Here we use the geometrical properties of the domain, i.e., that the space is free of obstacles and sensing uncertainty is described by a disk of fixed radius, together with the fact that I is a singleton. This will allow us to employ the construction used in Theorem 2.

Assume $\mathcal{I}(\kappa), (q, r) \models \exists_{\boxdot} \neg K_a \forall_{\Diamond} g$. This means that, in particular, $\mathcal{I}(\kappa), (q, r) \models \neg K_a \forall_{\Diamond} g$. That is, there exist a possible run R and time t such that $\mathsf{R}(t) = (q_1, r)$ and $\mathcal{I}(\kappa), (\mathsf{R}, t') \models \neg g$ for all $t' \leq t$. Denote the trajectory of the robot in R by Q. Hence, the portion of Q up to and including q_1 never passes through the goal. We can also conclude that there exists a sensing history h that can be attained along Q such that $h(t) = r$ and $h(t') = Q(t')$ for every $t' < t$.

We now show that there exists some h' such that $\mathcal{I}_h(\kappa), (q, h') \models \exists_{\boxdot} \neg K_a \forall_{\Diamond} g$. This requires demonstrating the existence of a run R' that reaches position q in such a way that the readings along this run are those described by h', and such that $\neg K_a \forall_{\Diamond} g$ is satisfied at each point of R' prior to (q, h'). However, we know by construction that every run with h as its readings satisfies $\neg K_a \forall_{\Diamond} g$ for all times $t' \leq t$, since these readings are obtained in R which up to time t does not pass through G. In fact, any run with sensing history that is obtained from h by rescaling the path parameter of h will have this property as well. (Recall that h up to t is in fact a continuous curve.) It remains to show that there is a run passing through q whose readings are described by such a history. That is, we must show that there is a trajectory Q_1 such that, for all $t' < t$, $|Q_1(t') - \hat{Q}(t')| \leq \rho$ (where $\hat{Q}$ is some trajectory obtained by rescaling Q) and $|dQ_1(t') - D| \leq \delta$, that $Q_1(t) = q$ and that

Q_1 is continuous and piecewise differentiable. The construction used in Theorem 2 provides a run with precisely these properties. □

PROOF OF THEOREM 4. We can now proceed to prove Theorem 4. Our strategy is as follows. Assume that a sound and complete termination condition depending on full sensing history exists. Using Theorem 3 we can assume this condition to be $K_a\forall_\diamond g$. We will show that the corresponding condition $K_a\forall_\diamond g$ in the system $\mathcal{I}(\kappa)$, which depends only on the current position reading, is also sound and complete.

Soundness Let R be a run. First, assume that there is a first time t in R in which $K_a\forall_\diamond g$ is satisfied w.r.t. $\mathcal{I}(\kappa)$. Denote the position at this time by q_0. If t is also the first time that $K_a\forall_\diamond g$ is satisfied in R w.r.t. $\mathcal{I}_h(\kappa)$, then because that condition is sound w.r.t that system, we know that g holds.

Suppose that t is not the first time in R in which $K_a\forall_\diamond g$ is satisfied w.r.t. $\mathcal{I}_h(\kappa)$. We will proceed as follows: we will show that for every $t' < t$ there is some $t' \leq t'' < t$ such that $(\mathsf{R}, t'') \models g$. This together with the fact that G is closed, implies that $q_0 \in G$.

Here is a formal argument. For every $t' < t$, we know that $K_a\forall_\diamond g$ is not satisfied in R at time t' w.r.t. $\mathcal{I}(\kappa)$. Hence, by Claim 1, for every point q along the run appearing at a time $t' < t$, we can construct a run R_q that passes through q and from q continues as R, and a history h_q that depends on q, such that $\exists_\boxminus \neg K_a\forall_\diamond g$ will be satisfied at (q, h_q) w.r.t. $\mathcal{I}_h(\kappa)$. Since $K_a\forall_\diamond g$ is sound and complete w.r.t. $\mathcal{I}_h(\kappa)$, we conclude that somewhere between t' (where R passes through q) and t (where R passes through q_0), g must be satisfied. To see this, recall that if $K_a\forall_\diamond g$ is satisfied w.r.t. $\mathcal{I}(\kappa)$ it is satisfied w.r.t. $\mathcal{I}_h(\kappa)$ as well. Hence, $K_a\forall_\diamond g$ is satisfied w.r.t. $\mathcal{I}_h(\kappa)$ at time t as well. Because $K_a\forall_\diamond g$ is sound w.r.t. to $\mathcal{I}_h(\kappa)$, $K_a\forall_\diamond g \wedge g$ must hold somewhere between t' and t in R_q w.r.t. $\mathcal{I}_h(\kappa)$. Because we can do this for every point arbitrarily close to q_0 and because G is closed, we conclude that q_0 is in G.

Next, assume that $K_a\forall_\diamond g$ is not satisfied at time t in R w.r.t. to $\mathcal{I}(\kappa)$. (Recall that t is the infimum of the set of time points in which $K_a\forall_\diamond g$ is satisfied w.r.t. $\mathcal{I}(\kappa)$). Using Claim 1, we know that there is a history h_{q_0} such that $(q, h_{q_o}) \models \exists_\boxminus \neg K_a(\forall_\diamond g)$. Thus, there is a run along which $K_a\forall_\diamond g$ is never satisfied w.r.t. $\mathcal{I}_h(\kappa)$ before q_0 (q_0 included). W.l.o.g., let R be that run. We know that t is also the infimum of time points in R in which $K_a\forall_\diamond g$ is satisfied w.r.t. $\mathcal{I}_h(\kappa)$. (Since it is satisfied w.r.t. to $\mathcal{I}_h(\kappa)$ whenever it is satisfied w.r.t. $\mathcal{I}(\kappa)$). We claim that because $K_a\forall_\diamond g$ is sound w.r.t. $\mathcal{I}_h(\kappa)$, it must be the case that there is some $\epsilon > 0$ such that g is satisfied in R for all times in $[t, t+\epsilon]$. Before we prove this claim, notice that it implies that $K_a\forall_\diamond g$ is sound w.r.t. $\mathcal{I}(\kappa)$ as well. If $K_a\forall_\diamond g$ is satisfied at some time t' in R, there must be some time t'' such that $t'' < t'$ and $t'' \in [t, t+\epsilon]$, at which $K_a\forall_\diamond g$ is also satisfied (since t is the infimum of time points in which $K_a\forall_\diamond g$ is satisfied) and by our claim, g is satisfied there as well.

It remains to show the following: If G is described by a finite collection of algebraic curves and the robot's path is described by an algebraic curve then (using the notations above) there is some $\epsilon > 0$ such that g is satisfied in R for all times in $[t, t+\epsilon]$. This follows from the property that the intersection of two such geometric objects can contain only a finite number of connected components and the fact that there is an infinite sequence of points in G whose limit is the position in R at time t.

Completeness The goal G is a compact subset of $\mathbb{R}^n$, which implies that it is a bounded subset of $\mathbb{R}^n$. Because $\delta < \pi/2$, and there is a progress guarantee, it follows that eventually, the robot will get to a point where it is beyond the goal, and cannot return to the goal. Notice that because there is a complete termination condition for depending on sensing history, we know that any trajectory will have to pass through the goal. Hence, once the robot reaches a position that is beyond the goal, and whose distance to the goal greater than ρ, the sentence $K_a \forall_{\diamond} g$ will necessarily hold even in $\mathcal{I}(\kappa)$. Hence, this termination condition is complete. $\square$

THEOREM 5. *The condition* $\mathsf{T}g$ *is an optimal sound termination condition.*

PROOF. First, note that $\mathcal{I}(\kappa) \models \mathsf{T}g \leftrightarrow K_a\mathsf{T}g$, thus $\mathsf{T}g$'s satisfaction depends only on the local state of the robot, and it does indeed represent a termination condition. By definition, we have that $\mathsf{T}g$ is sound because whenever it is satisfied, $K_a(g \vee \forall_{\diamond}(\mathsf{T}g \wedge g))$ is satisfied as well, which implies $g \vee \forall_{\diamond}(\mathsf{T}g \wedge g)$, i.e. either g holds or $\mathsf{T}g \wedge g$ holds earlier.

Next, we would like to show that $\mathsf{T}g$ is as strong as any other sound termination condition. Notice that the soundness proof above showed that every fixpoint of $X \leftrightarrow K_a(g \vee \forall_{\diamond}(X \wedge g))$ defines a sound termination condition. Let T be the largest (w.r.t. set inclusion) sound termination condition. (From Lemma 1, we know that a largest such termination condition exists.) Thus, T is an optimal sound termination condition. We will show that $\mathcal{I}(\kappa) \models T \leftrightarrow K_a(g \vee \forall_{\diamond}(T \wedge g))$, which implies that T is a fixpoint of this equation. Since $\mathsf{T}g$ is the greatest fixpoint of this equation and is sound, they must be equal. First, we show that $\mathcal{I}(\kappa) \models T \rightarrow K_a(g \vee \forall_{\diamond}(T \wedge g))$. Suppose that $\mathcal{I}(\kappa), s \models T$. Because T is sound we have that $\mathcal{I}(\kappa), s \models g \vee \forall_{\diamond}(T \wedge g)$. However, $\mathcal{I}(\kappa) \models T \rightarrow K_aT$, hence, $\mathcal{I}(\kappa) \models T \rightarrow K_a(g \vee \forall_{\diamond}(T \wedge g))$.

We must now show that $\mathcal{I}(\kappa) \models K_a(g \vee \forall_{\diamond}(T \wedge g)) \rightarrow T$. First, we notice that $K_a(g \vee \forall_{\diamond}(T \wedge g))$ is sound. To see this let R be a run and s a state in R satisfying $K_a(g \vee \forall_{\diamond}(T \wedge g)$. If $\mathcal{I}(\kappa), s \not\models g$ then $\mathcal{I}(\kappa), s \models \forall_{\diamond}(T \wedge g)$ which implies that $T \wedge g$ was satisfied in the past. Since we have established that, $\mathcal{I}(\kappa) \models T \rightarrow K_a(g \vee \forall_{\diamond}(T \wedge g))$, we get that if $K_a(g \vee \forall_{\diamond}(T \wedge g))$ is satisfied at some point, then either g holds at that point, or previously, $K_a(g \vee \forall_{\diamond}(T \wedge g)) \wedge g$ held. Hence, $K_a(g \vee \forall_{\diamond}(T \wedge g)) \wedge g$ is indeed sound. But T was constructed so that every other sound termination condition implies T. In particular, $\mathcal{I}(\kappa) \models K_a(g \vee \forall_{\diamond}(T \wedge g)) \rightarrow T$. $\square$

THEOREM 6. *Let* $\mathcal{I}$ *be a system. Define* $\mathsf{T}_\sigma \varphi$ *as follows:*

—$\mathsf{T}_0 \varphi \Leftrightarrow$ `true`;
—$\mathsf{T}_{\sigma+1} \varphi \Leftrightarrow K_a(\varphi \vee \forall_{\diamond}(\mathsf{T}_\sigma \varphi \wedge \varphi))$ *for successor ordinals;*
—$\mathsf{T}_\sigma \varphi \Leftrightarrow \bigwedge_{\tau<\sigma} \mathsf{T}_\tau \varphi$ *for limit ordinals.*

There exists an ordinal σ *for which* $\mathcal{I} \models \mathsf{T}\varphi \Leftrightarrow \mathsf{T}_\sigma \varphi$.

PROOF. Based on standard results in the theory of fixpoints, we know that if the operator used is monotone, the iterative method, initialized at `true` converges to the greatest fixpoint after a number of steps that is smaller than the cardinality of the set. Hence, it remains to show that if $\mathcal{I} \models \psi_1 \rightarrow \psi_2$ then $\mathcal{I} \models K_a(\varphi \vee \forall_{\diamond}(\psi_1 \wedge \varphi)) \rightarrow K_a(\varphi \vee \forall_{\diamond}(\psi_2 \wedge \varphi))$.

In order to see this, notice that $\mathcal{I} \models \psi_1 \to \psi_2$ implies that $\mathcal{I} \models (\psi_1 \wedge \varphi) \to (\psi_2 \wedge \varphi)$, which implies $\mathcal{I} \models \forall_{\diamond}(\psi_1 \wedge \varphi) \to \forall_{\diamond}(\psi_2 \wedge \varphi)$. However, we also know that $\mathcal{I} \models \psi_1 \to \psi_2$ implies that $\mathcal{I} \models (\varphi \vee \psi_1) \to (\varphi \vee \psi_2)$ and that $\mathcal{I} \models \psi_1 \to \psi_2$ implies $\mathcal{I} \models K_a\psi_1 \to K_a\psi_2$. Combining these results we obtain that $\mathcal{I} \models \psi_1 \to \psi_2$ implies $\mathcal{I} \models K_a(\varphi \vee \forall_{\diamond}(\psi_1 \wedge \varphi)) \to K_a(\varphi \vee \forall_{\diamond}(\psi_2 \wedge \varphi))$. $\square$

C. PROOFS OF RESULTS IN SECTION 5

Recall that in this section we are interested only in systems $\mathcal{I}$ for all termination condition T and for all runs in $\mathcal{I}$, there is a first time in which the termination condition T is true. In particular this implies that a termination condition T is sound w.r.t. $\mathcal{I}$ iff for every run $\mathsf{R} \in \mathcal{I}$, the first time T holds in R, the robot's position is in G.

LEMMA 4. *(a) If T is sound w.r.t. $\mathcal{I}$ then $\mathcal{I}[T] \models T \to g$.*
(b) T is sound w.r.t. $\mathcal{I}$ if and only if $\mathcal{I}[T] \models T \to K_a g$.
(c) If T is an optimal sound termination condition, then $\mathcal{I}[T] \models T \Leftrightarrow K_a g$.

PROOF. (a) Suppose that T is sound w.r.t. $\mathcal{I}$ and G. If $\mathcal{I}[T], s \models T$ then there is a run in $\mathcal{I}$ in which s is the first state satisfying T. Since T is sound w.r.t. $\mathcal{I}$ we have that $\mathcal{I}, s \models g$. Since g depends only on the robot's position, and that is determined by s, we obtain $\mathcal{I}[T], s \models g$.

(b) Assume that T is sound w.r.t. $\mathcal{I}$. From (a) we have that $\mathcal{I}[T], s \models T \to g$. However, since T is a condition on the robot's local state, we have that $\mathcal{I} \models T \Leftrightarrow K_a T$. Hence, $\mathcal{I}[T], s \models T \to K_a g$.

For the other direction, assume that $\mathcal{I}[T] \models T \to K_a g$. Notice that $\mathcal{I}[T]$ contains all states of $\mathcal{I}$ at which T holds for the first time in some run of $\mathcal{I}$. Since $\models K_a g \to g$, it follows that at the first time at which T holds in every run of $\mathcal{I}$, the robot is in g. Since, by assumption, for every run R of $\mathcal{I}$ in which T holds, there is a first time at which T holds in R, it follows that T is sound with respect to $\mathcal{I}$ and G.

(c) Because T is sound, we have from Part (b) that $\mathcal{I}[T] \models T \to K_a g$.

Now, suppose that $\mathcal{I}[T], s \models K_a g \wedge \neg T$. We now show that this implies the existence of a sound termination condition T' that is strictly stronger than T with respect to $\mathcal{I}$. This will contradict the optimality of T, and thus complete the proof.

Let s_a be the robot's local state in s. We define T' to be $T \cup \{s_a\}$, by which we mean that T' holds exactly if either T does, or the robot's local state is s_a. We now argue that T' is sound with respect to $\mathcal{I}$. Let R and t be such that T' holds at $\mathsf{R}(t)$. If T holds at $\mathsf{R}(t)$ then the soundness of T implies $\mathcal{I}, \mathsf{R}(t) \models \forall_{\diamond}(T \wedge g)$, and hence also $\mathcal{I}, \mathsf{R}(t) \models \forall_{\diamond}(T' \wedge g)$. If T doesn't hold at $\mathsf{R}(t)$, then the robot's local state at $\mathsf{R}(t)$ is s_a. We claim that in this case as well, we have $\mathcal{I}, \mathsf{R}(t) \models \forall_{\diamond}(T' \wedge g)$. For each point (R', t') at which the local state is s_a, we are in one of two possibilities. Either (i) T holds in R' at some time $t'' \leq t'$, in which case we have by the soundness of T that $T \wedge g$ (and hence also $T' \wedge g$) holds at some earlier point, or (ii) T does not hold in R' before time t'. In this case, by definition of $\mathcal{I}[T]$, the state $s' = \mathsf{R}'(t')$ appears in $\mathcal{I}[T]$. Since, by assumption, we have that $\mathcal{I}[T], s \models K_a g$ and the robot has the same local state in s and in s', it follows that $\mathcal{I}[T], s' \models g$, and since T' holds at s' we obtain that $\mathcal{I}[T], s' \models (T' \wedge g)$. Since in both cases (i) and (ii) we are guaranteed that at some earlier point $(T' \wedge g)$ must hold, we conclude that $\mathcal{I}, \mathsf{R}(t) \models \forall_{\diamond}(T' \wedge g)$. Thus, given Lemma 2 we have that T' is sound.

It remains to show that T' is strictly stronger than T. First, since $T' \stackrel{\mathrm{def}}{=} T \cup \{s_a\}$, it is clear that T' is at least as strong as T. To show that the opposite is not the case, we must show that $\mathcal{I} \not\models T' \rightarrow \forall_{\diamond} T$, i.e., that $\mathcal{I} \models T' \wedge \neg\forall_{\diamond} T$. That is, we must show that there is some run $\mathsf{R}' \in \mathcal{I}$ and some time t' such that T' holds at $\mathsf{R}'(t')$, but for no $t'' < t$ is it the case that T holds at $\mathsf{R}'(t'')$. However, we started this proof by assuming the existence of a state s such that $\mathcal{I}[T], s \models K_a g \wedge \neg T$. Consequently, based on the definition of T', we know that $\mathcal{I}[T], s \models T' \wedge \neg T$. Recalling the definition of $\mathcal{I}[T]$, we immediately conclude that a run $\mathsf{R}' \in \mathcal{I}$ and a time t' exist, such that $\mathsf{R}'(t') = s$ and $\mathcal{I}, s \models T' \wedge \neg\forall_{\diamond} T$. Hence, T' is strictly stronger than T and we are done. □

LEMMA 5. *Given G and κ, if the program $\mathsf{P_g}$ (defined as:* **if** $K_a g$ **then** halt*) has a unique implementation in the context κ, then the condition defined by $K_a g$ in the system $\mathcal{I}_g$ representing $\mathsf{P_g}$ in the context κ is an optimal sound termination condition for $\mathcal{I}(\kappa)$ and G.*

PROOF. Assume that the program $\mathsf{P_g}$ has a unique implementation in the context κ. Let T be the optimal sound termination condition in the context κ. Let $\mathcal{I}(\kappa)$ be the (uninhibited) interpreted system in that context. By Lemma 4 we have that $\mathcal{I}(\kappa)[T] \models T \Leftrightarrow K_a g$. This means that $\mathcal{I}(\kappa)[T]$ is a system representing P_g in this context. It follows that if $\mathsf{P_g}$ has a unique implementation in the context κ, then T is the unique implementation of $\mathsf{P_g}$ in this context. As $\mathcal{I}(\kappa)[T] \models T \Leftrightarrow K_a g$, we obtain that the condition defined by $K_a g$ in this system is an optimal sound termination condition for $\mathcal{I}(\kappa)$ and we are done. □

ACKNOWLEDGMENTS

We wish to thank Alvaro del Val, Nir Friedman, Ronny Kohavi, and Moshe Tennenholtz for important discussions concerning this work.

Special thanks to the anonymous referees for a very thorough and helpful review and to Joe Halpern for important comments on previous drafts.

Ronen Brafman and Yoav Shoham are supported in part by ARPA and AFOSR grants F49620-92-J-0547 and AF F49620-94-1-0090; Jean-Claude Latombe is supported in part by AFOSR grant AF F49620-94-1-0090; Yoram Moses is supported in part by a Helen and Milton A. Kimmelman Career Development Chair.

References

BRAFMAN, R. I., LATOMBE, J. C., MOSES, Y., AND SHOHAM, Y. 1994. Knowledge as a tool in motion planning under uncertainty. In R. Fagin, Ed., *TARK, Proc. of the 5th Conf.* (1994). Morgan Kaufmann, pp. 208–224.

BRAFMAN, R. I., LATOMBE, J. C., AND SHOHAM, Y. 1993. Towards knowledge-level analysis of motion planning. In *Proc. of the Eleventh National Conf. on AI* (1993). pp. 670–675.

BRAFMAN, R. I. AND SHOHAM, Y. 1995. Knowledge considerations in robotics and distribution of robotic tasks. In *Proc. Fourteenth International Joint Conference on AI* (1995).

CANNY, J. F. 1989. On computability of fine motion plans. In *Proc. of the 1989 IEEE Int. Conf. on Rob. and Aut.* (1989). pp. 177–182.

DONALD, B. R. 1989. *Error detection and recovery in robotics.* Springer-Verlag, Berlin.

DONALD, B. R. 1994. On information invariants in robotics. *Artif. Intell. 72*, 1, 217–304.

DWORK, C. AND MOSES, Y. 1990. Knowledge and common knowledge in a byzantine environment: crash failures. *Information and Computation 88*, 2, 156–186.

EMERSON, E. A. AND HALPERN, J. Y. 1985. Decision procedures and expressivenessin the temporal logic of branching time. *J. of Comp. and Sys. Sci. 30*, 1, 1–24.

ERDMANN, M. 1986. Using backprojection for fine motion planning with uncertainty. *Int. J. of Robotics Research 5*, 1, 19–45.

ERDMANN, M. 1994. Understanding actions and sensing by designing action-based sensors. In *IEEE ICRA Workshop on Design* (1994).

FAGIN, R., HALPERN, J. Y., MOSES, Y., AND VARDI, M. Y. 1995. *Reasoning about Knowledge.* MIT Press.

HALPERN, J. Y. AND FAGIN, R. 1989. Modelling knowledge and action in distributed systems. *Distributed Computing 3*, 159–177.

HALPERN, J. Y. AND MOSES, Y. 1990. Knowledge and common knowledge in a distributed environment. *J. ACM 37(3)*, 549–587.

HALPERN, J. Y. AND VARDI, M. Y. 1986. The complexity of reasoning about knowledge and time. In *Proc. 18th ACM Symp. on Theory of Computation* (1986). pp. 304–315.

HINTIKKA, J. 1962. *Knowledge and Belief.* Cornell University Press, Ithaca, NY.

KAELBLING, L. P. AND ROSENSCHEIN, S. J. 1990. Action and planning in embedded agents. *Robotics and Autonomous Systems 6*, 35–48.

LATOMBE, J. C. 1991. *Robot Motion Planning.* Kluwer Academinc Publishers, Boston.

LATOMBE, J. C., LAZANAS, A., AND SHEKHAR, S. 1991. Robot motion planning with uncertainty in control and sensing. *Artificial Intelligence 52*, 1–47.

LAZANAS, A. 1994. Reasoning about uncertainty in robot motion planning. Ph. D. thesis, Stanford University.

LEVESQUE, H. 1984. Foundations of a functional approach to knowledge representation. *Artificial Intelligence 23*, 2, 155–212.

LOZANO-PÉREZ, T. 1983. Spatial planning: a configuration space approach. *IEEE Trans. on Comp. C-32*, 2, 108–120.

LOZANO-PÉREZ, T., MASON, M. T., AND TAYLOR, R. H. 1984. Automatic synthesis of fine-motion strategies for robots. *Int. J. of Robotics Research 3*, 1, 3–24.

MASON, M. T. 1984. Automatic planning of fine motions: correectness and completeness. In *IEEE Int. Conf. on Rob. and Aut.* (1984).

MOORE, R. C. 1985. A formal theory of knowledge and action. In *Formal Theories of the Common Sense World* (1985).

MOSES, Y. AND TUTTLE, M. R. 1988. Programming simultaneous actions using common knowledge. *Algorithmica 3*, 121–169.

ROSENSCHEIN, S. J. 1985. Formal theories of knowledge in AI and robotics. *New Generation Comp. 3*, 345–357.

TARSKI, A. 1955. A lattice-theoretic fixpoint theorem and its applications. *Pacific Journal of Mathematics 5*, 285–309.

TAYLOR, R. H., MASON, M. T., AND GOLDBERG, K. Y. 1988. Sensor-based manipulation planning as a game with nature. In R. Bolles and B. Roth, Eds., *Robotics Research 4* (1988). MIT Press, pp. 421–429.

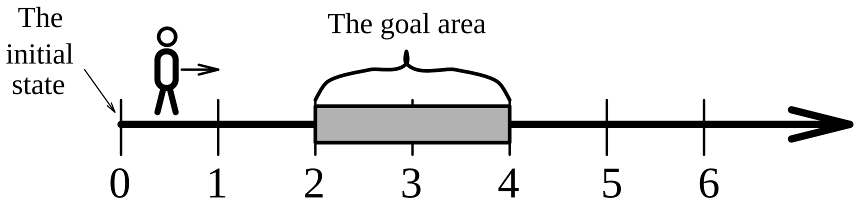

Fig. 1. A Simple Example

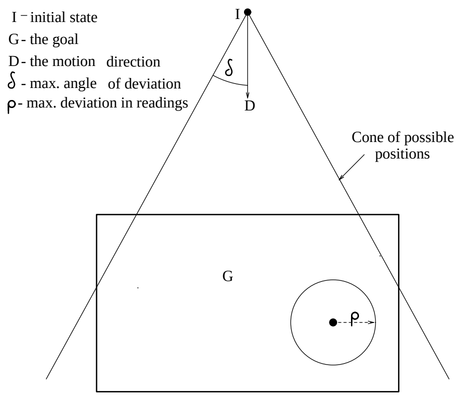

Fig. 2. A problem domain

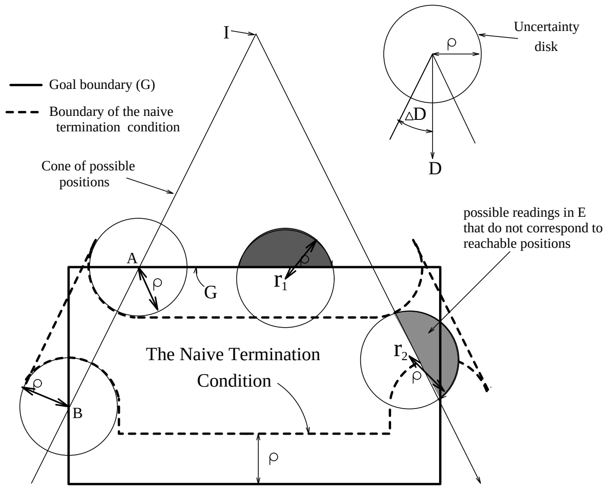

Fig. 3. A naive termination condition

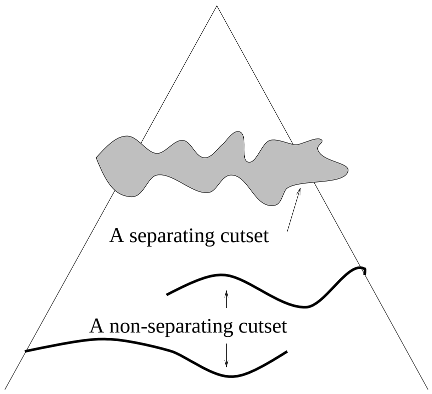

Fig. 4. Cutsets

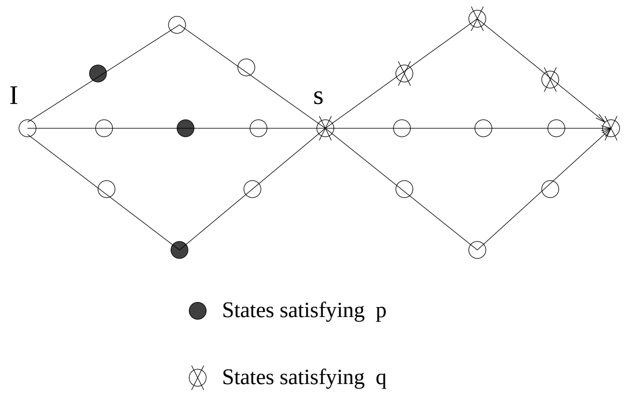

Fig. 5. Temporal Operators

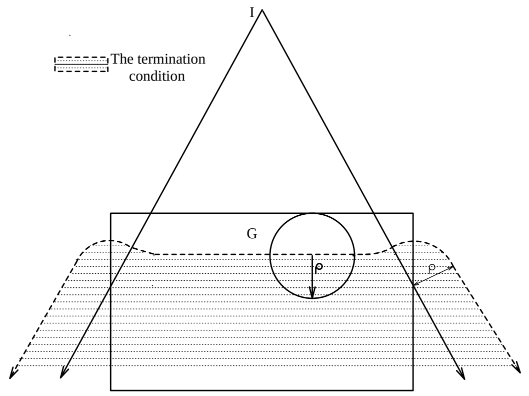

Fig. 6. A sound & complete termination condition

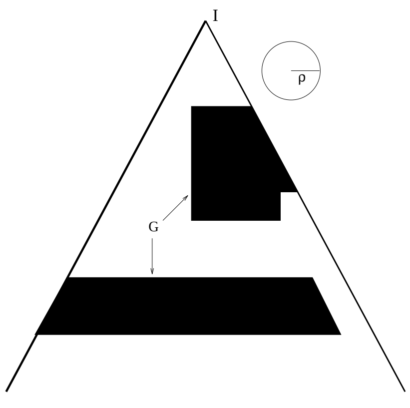

Fig. 7. The goal region

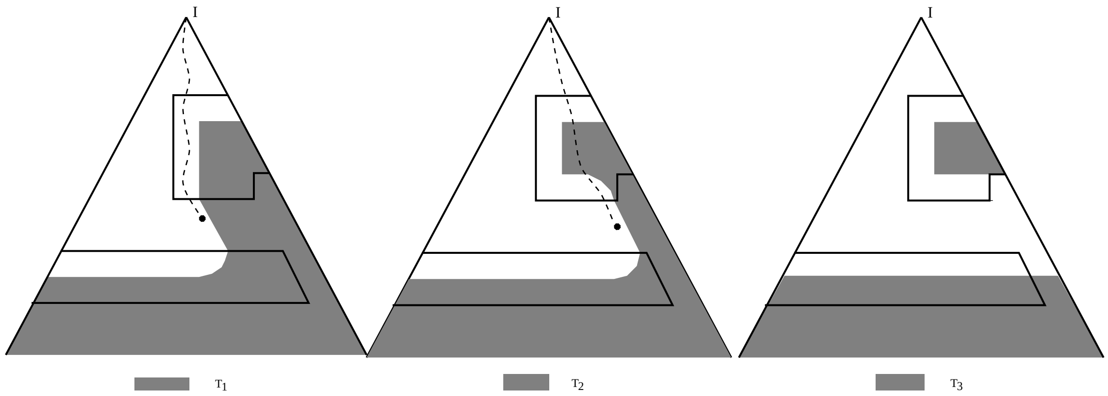

Fig. 8. T_1, T_2 and T_3