Limits of specifiability for sensor-based robotic planning tasks

Başak Sakçak Dylan A. Shell Jason M. O'Kane

***Abstract*— There is now a large body of techniques, many based on formal methods, for describing and realizing complex robotics tasks, including those involving a variety of rich goals and time-extended behavior. This paper explores the limits of what sorts of tasks are specifiable, examining how the precise grounding of specifications —that is, whether the specification is given in terms of the robot's states, its actions and observations, its knowledge, or some other information— is crucial to whether a given task can be specified. While prior work included some description of particular choices for this grounding, our contribution treats this aspect as a first-class citizen: we introduce notation to deal with a large class of problems, and examine how the grounding affects what tasks can be posed. The results demonstrate that certain classes of tasks are specifiable under different combinations of groundings.**

I. Introduction

Much work in robotics involves determining *how* to get robots to do things —choosing the steps to be taken, enacting them in sequence or under suitable of conditions. In concert, there is also distinct and growing body of work dedicated to expressing *what* we wish the robot to do, viz. describing the objectives of robot system. Common techniques include describing goal regions in state space, providing reward functions associated to states (or states and actions), and rich languages and logics that enable formal specifications of sequences. In the present paper, we examine the fundamental limits of these sorts of task descriptions. Such questions are essential but often overlooked: When one is unable to characterize the demands to be placed upon the robot, one should not expect the desired behavior to be elicited.

Figure 1 depicts an example of a robot moving through an environment, aided by a sensor that can detect landmarks in known locations. Such a robot might be tasked with localizing itself, though there is some subtlety in what exactly constitutes success: Must the robot come to know its own state and maintain that knowledge through the rest of its execution —in the language of temporal logic, 'eventually always' localized— or it is sufficient for the robot to know its own state periodically —'always eventually' localized? Of the two trajectories shown, both satisfy the latter requirement, but only Trajectory A satisfies the former. This simple example shows both the need for immaculate precision in task specification, lest the elicited behaviour mismatch the intent, and the potential for expressive representations like temporal logic to capture the relevant subtlety.

The first author is with the Center for Ubiquitous Computing, University of Oulu, Finland. The last two are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA. basak.sakcak@oulu.fi & {dshell, jokane}@tamu.edu. The first author was supported by Academy of Finland (project CHiMP 342556).

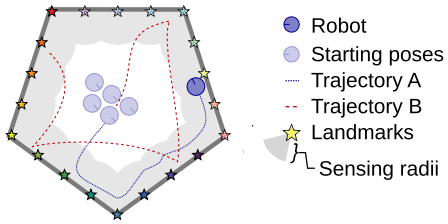

Fig. 1: A representative problem domain. Starting from an initial position with some uncertainty, a robot with imperfect motion dynamics navigates around an environment whose boundary contains landmarks that may be detected and distinguished from one another. The essence of the setting is the interplay of the robot's states, actions, and observations. This paper addresses the ability of various specification approaches to describe tasks such as localization, coverage, or navigation, in this sort of setting.

To that end, this paper introduces a formal structure in which questions about specifiability of tasks for robots may be understood in a precise way. This structure begins from the complete, infinite traces of a robot's interaction with its environment across time, including the evolution of the robot's states, along with the robot's actions (which influence this state evolution) and observations (which may partially reveal the state evolution to the robot). In this abstract sense, a task is simply a collection of such traces that correspond to completion of the objective.

The contribution of this paper is to provide a direct connection between this very broad concept of robotic tasks and more commonly-utilized specification languages such as linear temporal logic (LTL). We show that the precise set of atomic propositions used to ground such logical formulae impacts the tasks that can be specified, in direct and sometimes counter-intuitive ways. We consider tasks specified in terms of states, in terms of action/observation histories (generalizing LaValle's history information states [13] to infinite traces), and so-called information states which refer to sets of possible states. The results include a characterization of the existence or non-existence of tasks specifiable within various subsets of these three example specification types.

Following a concise review of related work (§II), the remainder of this paper introduces some preliminary definitions (§III), defines tasks and their posability (§IV) and specifications (§V). Upon that foundation, we prove several results about the limits of specifiability (§VI) before offering concluding remarks (§VII).

II. Related work

Despite a recent surge in interest to describe tasks for robots using primarily natural language [2], [11], [17], [18], [27], there has been sustained interest in a surprisingly wide

variety of forms of specification for many years. Careful study has been devoted to tasks specified as collections of partially-ordered rules that require adherence [6], contracts describing desired subsystem interactions [10], specifications on the narrative structure of events observed by the robot [21], blends of POMDPs and temporal logic [16], and domain specification generalization across classes of tasks [7], and even specifications borrowed from the study of human dance [14], [15].

From this wide array of options, one common choice for describing the desired behaviour of a robot —borrowed from the formal methods community— is LTL, a logic particularly suitable for specifying robot tasks with temporally extended goals. A large body of robotics research exists that relies on LTL specifications including within the context of mobile robot navigation [9], multi-agent systems [12], [26], and even to describe rules for changing the environment that the robot interacts with [4]. Other temporal logics have also been of interest, including signal temporal logic because of its suitability for continuous systems [19], [24].

There have been at least two major efforts to address questions about whether tasks of interest can be expressed within certain specification schemes. One approach is based on the study of various fragments of LTL and their ability to express certain classes of tasks [23], [25]. Another analyzed the expressivity of reward functions for prescribing desired behavior, including the *reward hypothesis* [1], [5]. This paper explores a different avenue; it focuses on the grounding of propositions that appear in a general class of specifications and connects them back to the basic descriptive model of the robot's interactions with the environment. Consequently, we discuss cases in which certain groundings prove insufficient to express a task of interest. In considering this grounding, our work shares a commonality with other work on varying levels of spatial abstraction in the propositions within a temporal logic specification [20]. Whereas other work's focus is on the synthesis of task specifications, our results establish conditions under which varying groundings are sufficient for expressing a particular task. Likewise, the results are orthogonal to other work that considers the case in which a duly specified task admits no correct plan for the robot to complete it [22].

III. Preliminaries

This section lays a foundation for our results by introducing a formal notion of sensor-based robot tasks.

First, some notational niceties. For a given set A, we write $A^{\mathbb{N}}$ to denote the set of all (one-sided) infinite sequences of elements of set A. For a function $g : A \to B$, $g^{-1}(b) \subseteq A$ is the preimage of $b \in B$ under g. Furthermore, for $C \subseteq A$, we will use $g(C)$ to denote the image of C under g. The powerset of A is denoted by $\wp(A)$. When referring to a sequence of tuples we will drop the parentheses whenever it is clear from the context. For example, a sequence of pairs $((a_1, b_1), (a_2, b_2), \dots) \in (A \times B)^{\mathbb{N}}$ will be equivalently denoted $(a_1, b_1, a_2, b_2, \dots)$.

A. Robot transition systems and complete traces

Our treatment's cornerstone is the robot transition system:

Definition 1 (Robot transition system): *A* robot transition system *is a tuple* $R = (X, U, f, h, Y, X_0)$ *consisting of*

- *Sets of states* X*, actions* U*, and observations* Y*.*
- *A transition function* $f : X \times U \to \wp(X) \setminus \{\varnothing\}$*, where* $f(x, u)$ *is the set of possible states that can be reached when action* u *is taken by the robot in state* x*.*
- *An observation function* $h : X \times U \times X \to \wp(Y) \setminus \{\varnothing\}$*, where* $y \in h(x, u, x')$ *is an observation the robot receives after taking action* u *in state* x *and arriving in state* x'*.*
- *A set of initial states* $X_0 \subseteq X$*.*

The idea is that our robot interacts with its environment, executing actions and receiving observations. This model can express both nondeterminism in its state transitions —from state x executing action u, the next state x' can be any of the states in $f(x, u)$— and nondeterminism in the sensor information available to the robot —when transitioning from x to x' by executing action u, the observation y received by our robot may be any of the observations in $h(x, u, x')$.

Example 1 (Robot transition system): Recall the example in Figure 1. To model this scenario as a robot transition system, let $X = E \times S^1$, in which $E \subseteq \mathbb{R}^2$ is the environment and S^1 represents orientation. For its actions, the robot can rotate to change its heading and then translate forward in each step, so its action space is $U = S^1 \times (0, d]$ for some maximum translation $d \in \mathbb{R}_+$. The state transition function f returns the results of this requested motion, including some inexactitude in the movements. This robot's sensing may be modeled with the observation space $Y = \big(\{\bigstar, \bigstar, \dots, \bigstar\} \times \mathbb{R}^2 \times S^1\big) \cup \{\perp\}$, in which h reports the identity and relative pose of the nearest landmark, if any are within a given maximum sensing range, or $\perp$ if no landmarks are in range.

To ensure that tasks evolving over indefinite periods of time (e.g., maintenance tasks) can be modeled, we are interested in infinite traces that describe the robot's execution.

Definition 2 (Universe and complete trace): *A* complete trace *(or, just, a trace) for a robot transition system* $R = (X, U, f, h, Y, X_0)$ *is an infinite sequence of state, action, and observation triples of the form* $\omega = (x_1, u_1, y_1, x_2, u_2, y_2, \dots) \in (X \times U \times Y)^{\mathbb{N}}$ *that satisfy* $x_1 \in X_0$ *and respect* f *and* h *such that for all* $i = 1, 2, \dots$, $x_{i+1} \in f(x_i, u_i)$ *and* $y_i \in h(x_i, u_i, x_{i+1})$*. We denote by* Ω_R *the set of all complete traces for* R*, and* $\Omega_R \subseteq (X \times U \times Y)^{\mathbb{N}} =: \overline{\Omega}_R$*, the* universe of traces.

Such traces are 'complete' in their full detail of how the execution unfolds: all states, actions, and observations.

B. Condensers

A central idea in this paper is the importance of reasoning about what can be accomplished with only a limited view of the complete trace ω. Next, we make this precise.

Definition 3 (Condenser): *For a given robot transition system R and an arbitrary set A, a* condenser *for R to A is a function with domain Ω_R and codomain A.*

The next two examples illustrate the concept of a condenser by introducing important special cases.

Example 2 (Action/observation-trace): Let $\mathcal{H} = (U \times Y)^{\mathbb{N}}$ denote the *action/observation-trace space* of a given robot transition system R. The *action/observation-trace condenser* $c_{\mathcal{H}} : \Omega_R \to \mathcal{H}$ discards the states from each complete trace, leaving only the actions and observations, so that

$$(x_1, u_1, y_1, x_2, u_2, y_2, \dots) \mapsto (u_1, y_1, u_2, y_2, \dots).$$

This condenser captures the idea that a robot generally will not have direct information about its states, but will have access to history of actions executed and observations received. The concept is closely akin to LaValle's history information space [13], but expressed over infinite traces.

Example 3 (State-trace): Let $\mathcal{X} = X^{\mathbb{N}}$ denote the *state-trace space* of a given robot transition system R. The *state-trace condenser* $c_{\mathcal{X}} : \Omega_R \to \mathcal{X}$ maps a complete trace to the respective state-trace by discarding the actions and the observations, such that

$$(x_1, u_1, y_1, x_2, u_2, y_2, \dots) \mapsto (x_1, x_2, \dots).$$

The two example condensers so far both have codomains composed of infinite traces, and both work by simply discarding information from the complete trace. This is a pattern that will recur, but is not a restriction.

C. Information spaces

Beyond $c_{\mathcal{H}}$ and $c_{\mathcal{X}}$, other condensers can be defined with codomain $\mathcal{I}^{\mathbb{N}}$, in which $\mathcal{I} = \wp(X)$ is called the *information space (I-space)* and its elements are referred to as *information states (I-states)*. A trace of I-states resulting from such a condenser is then called *an I-space trace*, usually denoted $(\iota_1, \iota_2, \dots)$. These condensers capture the possibility of uncertainty in the robot's state across time.

We are particularly interested in the following condenser, which generalizes the notion of nondeterministic I-spaces introduced by LaValle [13] to infinite sequences:

Example 4 (Nondeterministic I-state condenser): Let $F(x_i, u_i, y_i)$ denote the set of possible states at stage $i+1$ conditioned on the state, action, and observation at stage i:

$$F(x_i, u_i, y_i) = \{x' \in f(x_i, u_i) \mid y_i \in h(x_i, u_i, x')\}. \quad (1)$$

The *nondeterministic I-state condenser* $c_{\text{ndet}} : \Omega_R \to \mathcal{I}^{\mathbb{N}}$ maps a complete trace to an I-state–trace, that is,

$$(x_1, u_1, y_1, x_2, u_2, y_2, \dots) \mapsto (\iota_1, \iota_2, \dots)$$

such that, starting from $\iota_0 = X_0$, each subsequent ι_k is just

$$\iota_k = \bigcup_{x \in \iota_{k-1}} F(x, u_k, y_k). \quad (2)$$

IV. Tasks

Formally in our setting, a task can be viewed as simply a collection of complete traces, as defined next.

Definition 4 (Task): *A task T for a robot transition system R is a subset of Ω_R, that is, $T \subseteq \Omega_R$.*

A central objective of this paper is to understand the limits of posing and specifying tasks in ways that do not rely on the full detail of the complete traces. The condensers introduced in Section III-B provide a clean way to accomplish this. For a given condenser c, the question is whether a task of interest can be properly expressed in the codomain of c, or whether the information lost when applying c obscures some aspect relevant to the task. The next definition makes this distinction precise.

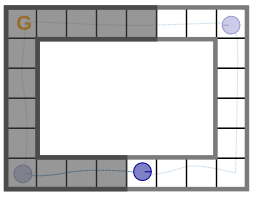

Fig. 2: Partly illuminated environment. The initial state is either bottom left facing east, or top right facing west (light blue circles). The goal is indicated by G.

Definition 5 (Well-posed under c): *A task $T \subseteq \Omega_R$ is* well-posed under condenser c *if $c^{-1}(c(T)) = T$. In particular, a task $T \subseteq \Omega_R$ is called*

1) state-trace posable *if T is well-posed under $c_{\mathcal{X}}$,*
2) action/observation-trace posable *if T is well-posed under $c_{\mathcal{H}}$, and*
3) I-state–trace posable *if T is well-posed under c_{ndet}.*

The intuition behind the notion of *well-posed* is, informally, that the codomain of c is 'rich enough' to describe the task properly, even in the absence of the details about each trace that are 'lost' when applying the condenser c.

In the following, to illustrate the notion of well-posedness under different condensers, we present examples building upon a common robot transition system. Refer to the environment in Figure 2 composed of an illuminated and dark cells that the robot can be in. The state space $X = E^{\Delta} \times \mathcal{O}$, in which E^{Δ} is the collection of grid cells and $\mathcal{O}$ are orientations corresponding to four cardinal directions. The set U of actions, which we assume to be deterministic, include rotating $90°$ in place and moving one step in the direction that the robot is facing. The start states X_0 and the goal region X_G are indicated in Figure 2, the latter corresponding to location G with any orientation. The robot is equipped with a light sensor with the set of observations $Y = \{\circ, \bullet, \circledcirc\}$ such that h reports either $\circ$ (light) or $\circledcirc$ (indeterminate) if the robot is in a well-lit cell and it reports $\bullet$ (dark) or $\circledcirc$ indeterminate otherwise.

Example 5 (State-trace poseable task): The task is to always keep re-visiting the goal region, resulting in the set

$$T = \{\omega \in \Omega_R \mid \text{elements of } X_G \text{ appear infinitely many times in } \omega\}. \quad (3)$$

For each $z = c_{\mathcal{X}}(w)$ satisfying $\omega \in T$, $c^{-1}(z) \subseteq T$ since any $\omega' \in \Omega_R$ with $c_{\mathcal{X}}(w') = z$ will be in T by construction. Furthermore, for each $\omega \in T$ there is at least one $z \in c_{\mathcal{X}}(T)$ such that $\omega \in c_{\mathcal{X}}^{-1}(z)$ since $c_{\mathcal{X}}$ is a function. Thus, $c_{\mathcal{X}}^{-1}(c_{\mathcal{X}}(T)) = T$ and T is state-trace poseable.

Example 6 (Action/observation-trace poseable task): The task is for the robot to resolve the ambiguity caused by its initialization and the geometry to determine which side of the environment it is on. Observations $\circ$ or $\bullet$, if

received, suffice to break the symmetry forever; therefore

$$T = \{\omega \in \Omega_R \mid \bigcirc \text{ or } \bullet \text{ appears somewhere in } \omega\ \}. \quad (4)$$

Following the same reasoning as in the previous example, one can show that T is well-posed under $c_{\mathcal{H}}$.

Example 7 (I-state–trace poseable task): Consider the task T defined in Equation (4). For any $\omega = (x_1, u_1, y_1, \dots x_n, u_n, y_n, \dots)$ with y_n being the first occurrence of an observation different than ◍, the respective I-state–trace $c_{\text{ndet}}(\omega) = (\iota_1, \dots, \iota_n, \dots)$ satisfies for all $i \geq n$ that ι_i is a singleton (the dark or the light cell consistent with the actions). Then, consider an I-state–trace $s \in c_{\text{ndet}}(w)$ with $\omega \in T$ and suppose there exists $\omega' \in c_{\text{ndet}}^{-1}(s)$ with $\omega' \notin T$. This cannot be possible since the I-state corresponding to the observation ○ or ● is a singleton indicating that an observation different than ◍ is received and that it uniquely identifies the observation (as ○/● is impossible from the symmetric state). Then, it is true for each $s \in c_{\text{ndet}}(T)$ that $c_{\text{ndet}}^{-1}(s) \subseteq T$ and that T is I-state–trace poseable. The interpretation of this task over I-state–traces is to have, at some point, the robot experience the ambiguity about its possible state vanish.

Example 8 (I-state–trace poseable task): Suppose now that, instead of the light detector, the robot has a goal detector such that $h(x, u, x') = 1$ if $x \in X_G$ and 0 otherwise. Define the task T as in Equation (3) but with Ω_R incorporating this new sensor. This has not affected the basis of the argument made in Example 5, so it remains state-trace poseable. Task T is also action/observation-trace poseable since $y_i = 1$ if and only if $x_i \in X_G$. Since $T \subseteq \Omega_R$ it respects f and h. Then, for any I-state–trace $(\iota_1, \iota_2, \dots) = c_{\text{ndet}}(\omega)$ with $\omega \in T$, it is always true that $\iota_i \subseteq X_G$ if $y_i = 1$ and $x_i \in X_G$, owing to Equation (1). The preimage $c_{\text{ndet}}^{-1}((\iota_1, \iota_2, \dots)) \subseteq T$ since having an $\omega' \in c_{\text{ndet}}^{-1}((\iota_1, \iota_2, \dots))$ with $\omega' \notin T$ implies either that $\omega' \notin \Omega_R$ or that c_{ndet} does not respect f and h, neither of which is true. Therefore, $c_{\text{ndet}}^{-1}(c_{\text{ndet}}(T)) = T$, and T is I-state–trace poseable.

Indeed, Example 8 hints at a general rule about I-state–trace posablity of a task that is state- and action/observation-trace poseable, which will be stated in Lemma 6. We will further analyze the relationships between different condensers in terms of task posablity in Section V-B.

V. Specifications

Definition 6 (Specification): *A* task specification σ *for robot* R *is some finite length description of a set* $\Gamma_\sigma \subseteq \overline{\Omega}_R$ *for which set membership can be readily computed. The task specified by* σ *is* $\Gamma_\sigma \cap \Omega_R$.

The reason behind the use of $\overline{\Omega}_R$ is that the constraints imposed by the robot system R need not be expressed by the specification. Those are, in some sense, obtained for free by running the robot. A specification's job is to delineate the boundary within the set Ω_R and is free to be arbitrary outside that set. Forcing a specification to capture the boundary of Ω_R within $\overline{\Omega}_R$ can be onerous and is, ultimately, unnecessary.

A. Specification technologies

Linear Temporal Logic (LTL) extends classical propositional logic through the addition of operators for time. A formula in LTL is a finite sequence of symbols expressed over a collection of Boolean propositions, PROP, which we take to be the set of atomic propositions. Then, the syntax of formulae in LTL is given recursively via the grammar

$$\phi \ ::= \ \top \mid \textbf{start} \mid \mathbf{q} \mid \neg\phi \mid \psi \vee \varphi \mid \bigcirc\phi \mid \psi \,\mathsf{U}\varphi,$$

where the ϕ, ψ, and φ are LTL formulae, $\top$ is the proposition representing Boolean value 'true', **start** is the temporal operator only satisfied at the initial time, atomic $\mathbf{q} \in$ PROP, $\neg$ denotes negation, $\vee$ disjunction, $\bigcirc$ the 'next' operator, and U the 'up until' operator. This allows derivation of $\bot \equiv \neg\top$, conjunction $(\phi \wedge \varphi) \equiv \neg(\neg\phi \vee \neg\varphi)$ via De Morgan's law, implication $(\phi \Rightarrow \psi) \equiv (\neg\phi \vee \psi)$, equivalence $(\phi \Leftrightarrow \psi) \equiv \big((\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)\big)$, and three additional temporal operators 'eventually' $\Diamond\phi \equiv (\top \,\mathsf{U}\phi)$, 'always' $\Box\phi \equiv (\neg\Diamond\neg\phi)$, and 'unless' $\psi \,\mathsf{W}\varphi \equiv (\psi \,\mathsf{U}\varphi \vee \Box\psi)$.

Semantics of LTL formulae is provided, like modal logics, via Kripke structures. The standard semantics is defined over infinite sequences: $s = (P_1, P_2, P_3, \dots)$ where $P_i \subseteq$ PROP are the propositions which hold (i.e., equal $\top$) at time i. A sequence s satisfies $\bigcirc\phi$ at time i when ϕ evaluates to $\top$ at time $i+1$. Sequence s satisfies $\psi \,\mathsf{U}\varphi$ at time i if there is some moment, either at i or later, for which φ holds, and ψ is $\top$ from i until the step just before then. We write $\langle s, i\rangle \models \psi$ if and only if the sequence s satisfies ψ at time i.

In all that follows, we will consider only robot transition systems where X, U, and Y are finite. This will mean PROP is finite as well. Mostly, we use a change in typeface to transfer from $X \sqcup U \sqcup Y$ to their associated propositions; we'll write $x \overset{\alpha}{\mapsto} \alpha(x) := \mathbf{x} \in$ PROP when an explicit association adds clarity (and similarly for actions or observations).

Definition 7: *Given state set* $X = \{x', x'', \dots, x^{(m)}\}$, *a* state-trace LTL formula *is a formula with* PROP $= \{\mathbf{x'}, \mathbf{x''}, \dots, \mathbf{x^{(m)}}\}$, *with each proposition* $\mathbf{x^{(i)}} = \alpha(x^{(i)})$, *and with semantics defined in the standard way over infinite sequences of sets of propositions, but grounded to sequences in* $\overline{\Omega}_R$ *via* $c_{X\text{PROP}} : \overline{\Omega}_R \to \wp(\text{PROP})^{\mathbb{N}}$*:*

$$(x_1, u_1, y_1, x_2, u_2, y_2, \dots) \mapsto (\{\mathbf{x_1}\}, \{\mathbf{x_2}\}, \dots).$$

Intuitively, we must bridge the two treatments to allow LTL with the usual semantics to represent what is usually modeled. The idea is that $c_{X\text{PROP}}$ connects the space of traces (of Definition 2) with propositional sequences by asserting that the proposition representing state at a given time holds when that is the state.

Lemma 1: *Every state-trace* LTL *formula expresses a state-trace posable task.*

Proof: The state-trace LTL formula ϕ describes the set $\Gamma_\phi = \{\omega \in \overline{\Omega}_R \,|\, \langle c_{X\text{PROP}}(\omega), \text{initial time}\rangle \models \phi\}$, and expresses the task $T_\phi = \Gamma_\phi \cap \Omega_R$. Any s satisfying ϕ, i.e., where $\langle s, \text{initial time}\rangle \models \phi$, that also comes from the image of $c_{X\text{PROP}}$, is a sequence of singleton proposition sets; there is an injection from any such sequence into $\mathcal{X}$. The collection of sequences in $\mathcal{X}$ obtained via this injection,

restricted to ωs originating within Ω_R, is just $c_{\mathcal{X}}(T_\phi)$. If $v \in c_{\mathcal{X}}^{-1} \circ c_{\mathcal{X}}(T_\phi)$ but $v \notin T_\phi$ then there must be some $w \in T_\phi$ where $c_{\mathcal{X}}(v) = c_{\mathcal{X}}(w)$. Since v and w agree on the state elements of the sequence, they can only differ on either actions or observations, or both. But any such a difference cannot cause $\langle c_{X\text{PROP}}(w), \text{initial time}\rangle \models \phi$ while $\langle c_{X\text{PROP}}(v), \text{initial time}\rangle \not\models \phi$, so $v \in \Gamma_\phi$. Since the domain of $c_{\mathcal{X}}$ is Ω_R, necessarily $v \in \Omega_R$ and, therefore, $v \in T_\phi$. This establishes that $c_{\mathcal{X}}^{-1} \circ c_{\mathcal{X}}(T_\phi) \subseteq T_\phi$. ■

Definition 8: *Given the action and observation sets* $U = \{u', u'', \ldots, u^{(m)}\}$ *and* $Y = \{y', y'', \ldots, y^{(n)}\}$, *an* action/observation-trace LTL formula *is a formula with* PROP $= \{\mathbf{u'}, \mathbf{u''}, \ldots, \mathbf{u^{(m)}}\} \sqcup \{\mathbf{y'}, \mathbf{y''}, \ldots, \mathbf{y^{(n)}}\}$, *with propositions* $\mathbf{u^{(i)}} = \alpha(u^{(i)})$ *and* $\mathbf{y^{(j)}} = \alpha(y^{(j)})$, *and with semantics defined in the standard way over infinite sequences of sets of propositions, but grounded to sequences in* $\overline{\Omega}_R$ *via* $c_{H\text{PROP}} : \overline{\Omega}_R \to \wp(\text{PROP})^{\mathbb{N}}$*:*

$$(x_1, u_1, y_1, x_2, u_2, y_2, \ldots) \mapsto (\{\mathbf{u_1}, \mathbf{y_1}\}, \{\mathbf{u_2}, \mathbf{y_2}\}, \ldots).$$

Unlike Definition 7, action/observation-trace formulae describe properties on sequences which most robots have direct access to under realistic assumptions. Such robots, for instance, must deal with ambiguity if sensors fail to totally resolve state; typically, robots must pick actions which exhibit a degree of robustness to uncertainty. The formulae of Definition 8 stipulate requirements directly on action/observation sequences; the area of sensor-based planning, not to mention reactive controllers, often perform activities to achieve tasks matching this form.

Lemma 2: *Every action/observation-trace* LTL *formula expresses an action/observation-trace posable task.*

Proof sketch: Proceed analogously to Lemma 1. □

Robots processing streams of actions and observations may treat uncertainty in a variety of ways. Some cases, albeit relatively few realistic ones, are 'certainty equivalent' [3, p. 76], so that some surrogate state, which although potentially not the actual state known by the system, may suffice to be used by the robot to reason about how to act. More generally, one may integrate data into an estimate describing what is known (or 'believed') from the history of actions and observations. In some instances this estimate, or even reduced descriptors derived therefrom, may be sufficient for the robot to act. Unsurprisingly, one may wish to specify tasks for the robot in terms of these estimates and this motivates the following.

Definition 9: *Given state set* $X = \{x', x'', \ldots, x^{(m)}\}$, *and an information space* $\mathcal{I} = \{\iota', \iota'', \ldots, \iota^{(m)}\}$, *an* I-state–trace LTL formula *is a formula with* PROP $= \{\mathbf{x'}, \mathbf{x''}, \ldots, \mathbf{x^{(m)}}\}$, *with each proposition* $\mathbf{x^{(i)}} = \alpha(x^{(i)})$ *being a plausible true state consistent with the data received so far, and with semantics defined in the standard way over infinite sequences of sets of propositions, but grounded to sequences in* $\overline{\Omega}_R$ *via* $c_{I\text{PROP}} : \overline{\Omega}_R \to \wp(\text{PROP})^{\mathbb{N}}$*:*

$$(x_1, u_1, y_1, x_2, u_2, y_2, \ldots) \mapsto (I_1, I_2, \ldots),$$

where $I_n := \{\mathbf{x'} \mid x' \in \iota_n\}$, *with* ι_n *from Equation* (2).

The reader may have expected a definition directly analogous to the previous two, i.e., over symbols $\mathbf{\iota_n} = \alpha(\iota_n)$.

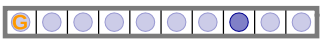

Fig. 3: A left/right moving robot, whose initial position is entirely unknown, occupies one of 10 cells in a corridor.

Such a definition is indeed possible, but the present definition has two merits: First, it is more economical in that LTL's semantics already permits sets of propositions to hold and, while the previous two definitions have not had cause to make use of this facility, the I-states do so naturally. Secondly, it emphasizes the vital conceptual point that two formulae written in terms of propositions corresponding to states (like $\mathbf{x'}$) will be indistinguishable at their surface level: whether the formulae represent statements about the ground-truth state or the robot's belief of a feasible state depends entirely on how the propositions are grounded to $\overline{\Omega}_R$.

Lemma 3: *Every I-state–trace* LTL *formula expresses an I-state–trace posable task.*

Proof sketch: Again, similar to that of Lemma 1. □

Corollary 1: *There are no state-, action/observation-, or I-state–trace* LTL *formulae for any tasks which are not state-, action/observation-, or I-state–trace posable, respectively.*

Proof sketch: Contrapositives of Lemmata 1, 2, 3. □

B. Relationships

Definition 10 (Equivalent LTL formulae): *Any two* LTL *formulae* ψ *and* ϕ *are* equivalent for robot system R *if and only if they specify the same task, i.e.,* $\Gamma_\psi \cap \Omega_R = \Gamma_\phi \cap \Omega_R$.

Example 9: Refer back to Example 8 which presented a task that was state-trace, action/observation-trace, and I-state–trace poseable. This task is also expressible in the respective domains with respective LTL formulae.

Lemma 4: *Any I-state–trace posable task is action/observation-trace posable.*

Proof: Given robot system R, for all $v, v' \in \Omega_R$, we will have that $c_{\mathcal{H}}(v) = c_{\mathcal{H}}(v') \implies c_{\text{ndet}}(v) = c_{\text{ndet}}(v')$ as the definition of c_{ndet}, via Equations (1) and (2), uses X_0 and then only the sequence of U and Y elements, i.e., it processes any two traces equal under $c_{\mathcal{H}}$ identically. ■

In its essence, the preceding shows that tasks posed in terms of I-states are no finer than those posable in terms of actions/observations. The phrase 'no finer' can be tightened to 'coarser' as the next example illustrates this can be strict.

Example 10: Consider the corridor environment in Figure 3. The state space is $X = \{1, 2, \ldots, 10\}$, and the robot can start anywhere, so $X_0 = X$. The robot is equipped with two actions: move Right or Left, which give $x \overset{\text{R}}{\mapsto} \max(x+1, 10)$ and $x \overset{\text{L}}{\mapsto} \min(1, x-1)$, respectively. Suppose that the robot has a sensor which returns the distance to the left-hand wall, but with a noisy sign, modeled via $h(x, u, x') = \{x', -x'\}$ and $Y = \{-10, \ldots, 10\} \setminus \{0\}$. Task T consists of those traces in which some non-positive sensor reading appears (i.e., the robot obtains concrete evidence of noise-induced mis-measurement). It is straightforward to specify in action/observation-trace LTL. But states (or even

sets of states) provide no ability to discriminate between receipt of $\pm n \in Y$, so the problem is neither state- nor I-state–trace poseable.

The following two examples will show that this 'ordering' intuition doesn't extend analogously to state-trace poseable (and LTL specified) tasks.

Example 11: Refer again to the task T in Example 6 that was shown to be action/observation and I-state–trace poseable. It is also expressible with respective LTL formulae. However, T is not state-trace poseable. To show this, consider $\omega \in T$ for which $\bigcirc$ appears only once and the respective state-trace $z = c_{\mathcal{X}}(\omega)$. The preimage $c_{\mathcal{X}}^{-1}(z)$ contains also the trace ω' with $\bullet$ instead, since that observation is certainly possible there as well. Therefore, $c_{\mathcal{X}}^{-1}(z) \not\subseteq T$ and T is not state-trace poseable. No state-trace LTL formula exists for this task, by Corollary 1.

Example 12: Revisit the environment and robot in Example 10, but now remove the robot's sensor. Here, consider a task T that is achieved if and only if the robot visits the G location at the far left of Figure 3. Then $T \subseteq \Omega_R$ is simply the collection of traces ω in which '1' appears somewhere in $c_{\mathcal{X}}(\omega)$. In state-trace LTL, it is specified as $\Diamond \mathtt{x_1}$. This task is neither action/observation-trace nor I-state–trace poseable: any sequences where, in any prefix, the total number of 'L' actions does not exceed the total number of 'R' actions so far appearing, will confuse initial locations 1 and 2. The former must be in T, the latter ought not to be.

Lemma 5: *Let ψ_x be a state-trace* LTL *formula specifying task T for robot transition system R. Then there is an action/observation-trace* LTL *formula $\psi_{u/y}$ expressing task T if it is action/observation-trace posable.*

Proof sketch: From $\mathcal{A}$, the Büchi automaton describing ψ_x, and system R, take a product to form nondeterministic Büchi automaton $\mathcal{A}'$, and establish $T = \Gamma_{\mathcal{A}} \cap \Omega_R = \Gamma_{\mathcal{A}'} \cap \Omega_R$. Next, it remains to show that $\mathcal{A}'$ corresponds to some formula $\psi_{u/y}$, as the class of Büchi automata is strictly larger than the class realized by LTL formulae. Diekert and Gastin provide *counter-freeness* [8, p. 286] as a characterization suitable of nondeterministic Büchi automata realizable via formulae. We establish that $\mathcal{A}'$ is counter-free by supposing the contrary, and showing this would cause $\mathcal{A}$ to not be counter-free. □

Remark 1: To collect the previous results: if the task T is action/observation-trace posable and one has a state-trace LTL formula specifying T (hence, via Lemma 1, it is state-trace posable) then Lemma 5 says there is an action/observation-trace LTL formula for T. Conversely, should T not be action/observation-trace posable, Corollary 1 indicates that no action/observation-trace LTL formula will specify it. As specifiability is narrower than posablity, it is interesting that non-specifiability allows conclusions about non-posablity:— with state-trace formula ψ_x specifying task $\hat{T}$, and knowledge that no $\psi_{u/y}$ specifies $\hat{T}$, then one can conclude task $\hat{T}$ is not action/observation-trace posable.

Corollary 2: *Let ψ_i be an I-state–trace* LTL *formula specifying task T for system R. Then, there is an action/observation-trace* LTL *formula $\psi_{u/y}$ expressing task T.*

Proof sketch: From $R = (X, U, f, h, Y, X_0)$ construct system $R' = (X', U, F', H', Y, \{X_0\})$, where $X' = \wp(X) \setminus \{\varnothing\}$, and, using Equation (1), $F'(A, u, y) = \bigcup_{x \in A} F(x, u, y)$ along with suitable H. Then, I-state–traces on R are state-traces on R', and T is action/observation-trace posable in R' if and only if it is in R. Next, transform ψ_i, written in terms of $\mathtt{x_i}$s, into ψ_i' over $\mathtt{\wr_j}$s through the (purely syntactic) rewriting. Lemma 5 implies some $\psi_{u/y}'$ exists provided the task is action/observation-trace posable for R or R'. Lemma 4 indicates it must be. □

Lemma 6: *Every task that is state-trace and action/observation-trace posable is also I-state–trace posable.*

Proof sketch: Suppose the contrary, then there must be a pair of sequences, $w \in T$ and $\bar{w} \notin T$ that look identical under c_{ndet}, but can be distinguished under $c_{\mathcal{H}}$ and $c_{\mathcal{X}}$. Consider a sequence w', with the states from w and the action/observations of $\bar{w}$. Sequence w' is in Ω_R because it generates the same I-state–trace as w and $\bar{w}$. But under $c_{\mathcal{H}}$, w' is in T, while under $c_{\mathcal{X}}$ w' is out. □

VI. Punchline: Limits of Specifiability

Consider the following family of claims regarding the limits of specifiability which may be constructed by resolving the expressible/inexpressible choices in the template below:

Claims 1–7: *There exist a robot transition system R (with finite X, U, Y) and some task T for which T is (in-)expressible in state-trace LTL formulae, (in-)expressible in action/observation-trace LTL formulae, and (in-)expressible in I-state–trace LTL formulae.*

Eagle-eyed readers will note that all seven of these claims have already been resolved in the preceding text. Here is a summary of results in table form.

State	Act./Obs.	I-state	Claim holds	Details
0	0	1	No	Corollary 2
0	1	0	Yes	Example 10
0	1	1	Yes	Example 11
1	0	0	Yes	Example 12
1	0	1	No	Corollary 2
1	1	0	No	Lemma 6
1	1	1	Yes	Example 9

VII. Conclusion and Outlook

The present contribution is fairly unusual as a robotics paper in that it deals with robot tasks without engaging in the question of how to identify optimal controls, select good actions, or find plans that allow a system to solve such tasks: the paper is concerned only with how one might characterize behavior satisfying the requirements of some task one might have in mind. It is declarative in its entirety. We do not discuss how a robot might perform actions to realize a task—indeed, the formalism being built on traces means, perhaps surprisingly, that no notion of the present or current time appears at all. It is our hope that this quizzical fact might stimulate fruitful discussion.

REFERENCES

[1] D. Abel, W. Dabney, A. Harutyunyan, M. K. Ho, M. Littman, D. Precup, and S. Singh, "On the expressivity of Markov reward," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7799–7812, 2021.

[2] J. Arkin, M. R. Walter, A. Boteanu, M. E. Napoli, H. Biggie, H. Kress-Gazit, and T. M. Howard, "Contextual awareness: Understanding monologic natural language instructions for autonomous robots," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication*, 2017.

[3] D. Bertsekas, *Reinforcement learning and optimal control*. Athena Scientific, 2019, vol. 1.

[4] L. Bobadilla, O. Sanchez, J. Czarnowski, K. Gossman, and S. M. LaValle, "Controlling wild bodies using linear temporal logic," in *Proc. Robotics: Science and Systems*, 2012.

[5] M. Bowling, J. D. Martin, D. Abel, and W. Dabney, "Settling the reward hypothesis," in *International Conference on Machine Learning*. PMLR, 2023, pp. 3003–3020.

[6] A. Censi, K. Slutsky, T. Wongpiromsarn, D. Yershov, S. Pendleton, J. Fu, and E. Frazzoli, "Liability, ethics, and culture-aware behavior specification using rulebooks," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8536–8542.

[7] A. Curtis, T. Silver, J. B. Tenenbaum, T. Lozano-Pérez, and L. Kaelbling, "Discovering state and action abstractions for generalized task and motion planning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 5, 2022, pp. 5377–5384.

[8] V. Diekert and P. Gastin, "First-order definable languages," in *Logic and Automata: History and Perspectives*, ser. Texts in Logic and Games, J. Flum, E. Grädel, and T. Wilke, Eds. Amsterdam: Amsterdam University Press, 2008, pp. 261–306.

[9] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic mobile robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[10] K. Ghasemi, S. Sadraddini, and C. Belta, "Compositional synthesis for linear systems via convex optimization of assume-guarantee contracts," *Automatica*, vol. 169, p. 111816, 2024.

[11] Z. Hu, F. Lucchetti, C. Schlesinger, Y. Saxena, A. Freeman, S. Modak, A. Guha, and J. Biswas, "Deploying and evaluating llms to program service mobile robots," *IEEE Robotics and Automation Letters*, 2024.

[12] Y. Kantaros, S. Kalluraya, Q. Jin, and G. J. Pappas, "Perception-based temporal logic planning in uncertain semantic maps," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2536–2556, 2022.

[13] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, also available at http://lavalle.pl/planning/.

[14] A. LaViers, Y. Chen, C. Belta, and M. Egerstedt, "Automatic Sequencing of Ballet Poses," *IEEE Robotics and Automation Magazine*, vol. 18, no. 3, pp. 87–95, 2011.

[15] A. Laviers and M. Egerstedt, "The ballet automaton: A formal model for human motion," in *Proc. American Control Conference*, 2011, pp. 3837–3842.

[16] J. Liu, E. Rosen, S. Zheng, S. Tellex, and G. Konidaris, "Leveraging Temporal Structure in Safety-Critical Task Specifications for POMDP Planning," in *Proc. Robotics: Science and Systems*, 2021.

[17] J. X. Liu, Z. Yang, I. Idrees, S. Liang, B. Schornstein, S. Tellex, and A. Shah, "Grounding complex natural language commands for temporal tasks in unseen environments," in *Conference on Robot Learning*. PMLR, 2023, pp. 1084–1110.

[18] J. X. Liu, Z. Yang, B. Schornstein, S. Liang, I. Idrees, S. Tellex, and A. Shah, "Lang2LTL: Translating Natural Language Commands to Temporal Specification with Large Language Models," in *Workshop on Language and Robotics at CoRL 2022*, 2022.

[19] W. Liu, W. Xiao, and C. Belta, "Learning robust and correct controllers from signal temporal logic specifications using barriernet," in *Conference on Decision and Control (CDC)*, 2023, pp. 7049–7054.

[20] Y. Oh, R. Patel, T. Nguyen, B. Huang, M. Berg, E. Pavlick, and S. Tellex, "Hierarchical planning with state abstractions for temporal task specifications," *Autonomous Robots*, vol. 46, no. 6, pp. 667–683, 2022.

[21] H. Rahmani, D. A. Shell, and J. M. O'Kane, "Planning to chronicle: Optimal policies for narrative observation of unpredictable events," *International Journal of Robotics Research*, vol. 42, no. 6, pp. 412–432, Mar. 2022.

[22] V. Raman and H. Kress-Gazit, "Analyzing unsynthesizable specifications for high-level robot behavior using LTLMoP," in *Computer Aided Verification (CAV)*. Snowbird, UT: Springer, Jul. 2011, pp. 663–668.

[23] R. Singh and I. Saha, "An Online Planning Framework for Multi-Robot Systems with LTL Specification," in *Proc. ACM/IEEE International Conference on Cyber-Physical Systems*, 2024.

[24] D. Sun, J. Chen, S. Mitra, and C. Fan, "Multi-agent motion planning from signal temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3451–3458, 2022.

[25] J. Tumova, A. Marzinotto, D. V. Dimarogonas, and D. Kragic, "Maximally satisfying LTL action planning," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[26] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus, "Optimality and robustness in multi-robot path planning with temporal logic constraints," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 889–911, 2013.

[27] R. Wang, Z. Yang, Z. Zhao, X. Tong, Z. Hong, and K. Qian, "LLM-based Robot Task Planning with Exceptional Handling for General Purpose Service Robots," *arXiv preprint arXiv:2405.15646*, 2024.