Specifying and achieving goals in open uncertain robot-manipulation domains

Leslie Pack Kaelbling
Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: lpk@mit.edu

Alex LaGrassa
Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: lagrassa@mit.edu

Tomás Lozano-Pérez
Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: tlp@mit.edu

***Abstract*—This paper describes an integrated solution to the problem of describing and interpreting goals for robots in open uncertain domains. Given a formal specification of a desired situation, in which objects are described only by their properties, general-purpose planning and reasoning tools are used to derive appropriate actions for a robot. These goals are carried out through an online combination of hierarchical planning, state-estimation, and execution that operates robustly in real robot domains with substantial occlusion and sensing error.**

I. Introduction

We would like to have intelligent robots that perform tasks in complex open environments such as homes, warehouses, and hospitals. As robots become more sophisticated, tasks can be specified using high-level goals, which the robot achieves by formulating and executing plans to move through, sense, and manipulate the world around it.

In such domains, the goals specified by humans for the robot are generally states of the world, rather than states of the robot, requiring some objects in the world (dishes, boxes, medicine bottles) to be in particular locations (a dishwasher, loading dock, or patient's table) or states (clean, taped shut, empty). It is critical to be able to specify such goals even when there is substantial uncertainty in the domain: it might be that neither the human nor the robot is aware of the location, state, or even existence of the particular objects needed when the goal is articulated.

Given such a goal, the robot might have to do significant work in the physical world just to be able to interpret it concretely; for example, the robot might need to search for appropriate objects or measure properties of known objects to see if they are suitable for the task. In this paper, we describe an integrated solution to the problem of describing, interpreting, and carrying out goals for robots in open uncertain domains. A critical feature of our approach to understanding the meaning of goal expressions is that it is carried out through the same planning, inference, and execution mechanisms as are used for determining physical robot actions. Thus, the system can use all of its physical abilities in service of gathering information in order to understand goal expressions in a way that will allow it to take physical actions to achieve the ultimate objective.

We show how to specify goals involving partially specified objects to a robot in an uncertain domain; provide inference rules and planning operators that can be used to augment an existing system for robot manipulation planning and execution under uncertainty so that it can act to interpret and achieve these goals; and demonstrate that the robot can actively interpret goals in open domains, by interacting with physical objects and searching the space around it, both in simulation and on a real physical robot, in the presence of substantial occlusion and sensing error. Figure 1 provides an illustration of the integrated system running on a PR2 robot.

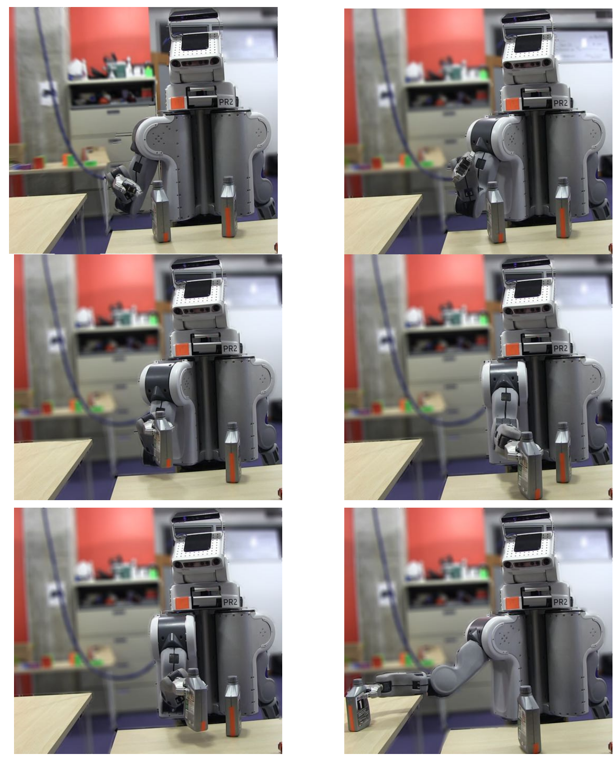

Fig. 1: Actions in response to the goal of having a heavy oil bottle on the table to the right. The robot picks up one bottle, feels that it is light, picks up the other one, and places it on the correct table, requiring re-planning and sensing.

Formally, this problem is a partially observed Markov decision process (POMDP), although it would be very difficult to formalize it as such, given fundamental uncertainty even about the dimensionality of the underlying state space. Our solution takes inspiration from online approximation strategies. Goals for the system are specified in *belief space* and the robot makes optimistic open-loop plans using a determinized model of the belief-space dynamics. After executing the first action of the plan, the robot obtains an observation, updates its belief (which may include reasoning about free space and the addition of newly postulated objects to its representation) and replans if necessary [16].

II. Related work

Much of the previous work in reference resolution for robots has focused on ambiguity in the utterance [5, 18, 10], including purely syntactic ambiguity as well as reference ambiguity. Tellex et al. [20] were one of the first groups to symbolically ground references in natural language to concrete representations in the world using probabilistic models. These approaches are largely passive, in the sense that they do not explicitly plan to gather disambiguating information.

Some planners can solve problems in open worlds, in which the robot does not know about all of the objects in advance. These methods do not explicitly model uncertainty in the planner, which limits their ability to handle noisy environments, but they do have the ability to select actions to gather information about unknown objects. For example, the planner used by Talamadupula et al. can satisfy quantifiable goals referring to unknown objects, like "a human," but the planner does not model uncertainty about properties specified by the goal. Furthermore, their approach does not account for actions that modify the state of the objects the robot is sensing, which is important for mobile manipulation domains. Replanning occurs when the robot discovers something new about the environment, like a new object, but not on plan failure [19], which makes the approach somewhat less robust than ours in noisy domains. The planner used by Joshi et al. [8] computes policies based on all possible maps, which is computationally expensive. It uses a reactive policy to avoid replanning, but must do so when a new object is discovered in the world.

One interesting thread of research uses dialog with humans to disambiguate references by detecting reference and world ambiguity [18, 7], and in some work additionally determining which questions would be useful to ask [3, 22, 4]. For example, Mavridis and Dong [13] combined sensing and clarification questions to resolve references with a system using single-step lookahead and discrete properties.

Our work focuses on the case where there is no uncertainty about the specification, but significant uncertainty about the domain. Planning to disambiguate the goal specification is handled by the same mechanism as planning to achieve goals more generally, allowing the robot to use all of its mobile manipulation capabilities in service of understanding and then achieving goals.

Another line of related work, in the robotics community, conceives of the problem in terms of symbol grounding and anchoring. The work that is closest in spirit to ours is that of Coradeschi and Saffiotti [2], which focuses on creating a mapping between a symbol system and objects in a perceptual system. They implement a system that can construct conditional plans with observation actions to find an appropriate mapping, but it is unable to address problems in which the plan involves objects that were not previously known. There are more modern extensions that have larger scope and more general perception [1, 21], and that handle complex natural language [12] but do not take physical actions to aid interpretation of instructions.

III. Denoting objects

In order for a robot to interpret a goal in an open world, it must have its own internal beliefs about the world state, have a language of expressions for objects, and have a way of evaluating expressions with respect to its current belief in order to determine which objects it is currently aware of, if any, are likely to satisfy the expression.

a) Belief representation and reasoning: We assume the robot has a representation of its belief about the world that is organized in terms of objects and probability distributions over their properties. Concretely, in our running example, there are rigid objects with distributions over properties:

- **type**: multinoulli distribution over a fixed finite set of possible object types;
- **pose**: objects are assumed to be resting on a stable face, so the pose has four degrees of freedom, (x, y, z, θ); we represent a joint distribution over all object poses, together with the robot's base pose, using a multivariate Gaussian in tangent space, which is updated using a variant of the unscented Kalman filter [6];
- **color**: truncated Gaussian in hue-saturation-value space;
- **weight**: Gaussian in log-weight space.

This representation is designed to support a high-fidelity belief-update step based on object detections from a 3D sensor for type and pose, RGB pixel values for color, and force-torque measurements from the wrist for weight.

In this representation, objects have no given names, unless they were specified for the robot in advance. So, for example, one might provide an initial belief describing, at least roughly, the positions of some known objects such as tables, and give them explicit names at initialization time. Any objects that the robot discovers as it is interacting with the domain, however, are added to the belief state with internal indices that have no external meaning, but that can be used as *anchors* to instantiate existentially quantified variables.

For the purposes of planning and reasoning, we characterize sets of detailed beliefs using a language of *belief fluents*, which use a type of epistemic operator to characterize the robot's belief about aspects of the world state. Let ϕ be a *random fluent*, representing a Boolean random variable, such as whether an object is contained in a region. Then we define

$$B_b(\phi, p) \equiv P_b(\phi = T) \geq p$$

to mean that the agent *believes* ϕ holds with probability at least p in the current belief state b, though we will generally suppress the b subscript for clarity. It is a fluent because during execution the underlying belief b will change, and so will the truth value of the belief fluent. Similarly, for a continuous random variable, such as the color of an object, we define

$$B(\phi, \mu, \Sigma, \Delta, p) \equiv |\mu - M_b| \leq \Delta \wedge S_b \preceq \Sigma \wedge P_b \geq p$$

where the belief distribution on quantity ϕ can be described by a Gaussian with parameters M_b, S_b that has mixture weight P_b, and where $\Sigma_1 \preceq \Sigma_2$ is a relation on covariance matrices that holds if the equi-probability contour of Σ_1 is contained in the equi-probability contour of Σ_2 for any fixed probability. This describes a set of probability distributions that are, in some sense, close in mean and at least as certain as a specified distribution.

b) Denoting expressions: Denoting expressions can be used to describe objects in terms of their properties without explicitly naming them. Because of uncertainty in the underlying properties of the objects, we can never be certain whether a denoting expression holds of a particular object; instead we will characterize the robot's belief using belief fluents of the form: $B(\text{DEN}(expr, obj), p)$, which means that the robot believes, with probability at least p, that obj can be denoted by $expr$, where obj is an internal name, or *anchor* for an object. These denoting expressions are *indefinite* and so it is possible that one might be true simultaneously for many different values of obj, or none at all. *Definite* descriptions, which imply the existence of a single satisfying object, are of great importance, but not handled in our current implementation.

We use a variation on classical lambda expressions of the form $\lambda X.expr$ where X is a variable that may occur in $expr$; legal expressions include conjunctions, disjunctions, and existential quantification.

The probability that random fluent $Den(\lambda X.expr, O)$ is true in a belief state b is computed by recursion on $expr$. Let σ be a *substitution* which maps variables into constant symbols. The application of a substitution to an expression replaces all free occurrences of each variable in σ with the associated constant. We will write $\sigma(expr)$ to stand for the expression that results from applying substitution σ to $expr$, and write substitutions in the notation of Python dictionaries.

We assume the ability to find the probability of a *ground* relational expression $R(c_1, \ldots, c_n)$ where $c_1, \ldots, c_n$ are numeric constants or anchors to objects, in b. So, for example, we would evaluate $Red(\_o34\_)$ by finding the distribution on the color of $\_o34\_$ in b and integrating the probability over the set of colors defined to be red.[1] We will write this quantity as $b(R(c_1, \ldots, c_n))$. The probability that the random fluent $Den(\lambda X.expr, O)$ is true is $\text{EVAL}(\{X : O\}(expr), b)$. Making strong independence assumptions, we define

- $\text{EVAL}(R(c_1, \ldots, c_n), b) = b(R(c_1, \ldots, c_n))$.
- $\text{EVAL}(expr_1 \wedge expr_2, b)$ $= \text{EVAL}(expr_1, b) \cdot \text{EVAL}(expr_2, b)$.
- $\text{EVAL}(expr_1 \vee expr_2, b) = \text{EVAL}(expr_1, b) +$ $\text{EVAL}(expr_2, b) - \text{EVAL}(expr_1, b) \cdot \text{EVAL}(expr_2, b)$.
- $\text{EVAL}(\exists X.expr, b) = \text{EVAL}(\bigvee_{o \in \mathcal{U}} \{X : o\}(expr))$ where $\mathcal{U}$ is the universe of objects.

[1]The semantics of color expressions in natural language is subtle and complex and we do not address it seriously here; we simply define color names as fixed volumes in HSV space.

c) Rigid designators: In order to plan in situations when it is not initially clear which objects will be used to satisfy a goal, we need to reason about whether the robot concretely knows which objects it needs to manipulate. We can, for example, call the $Place$ operator on any object that is represented in the belief state using its internal anchor as a name, but we cannot call it on $\lambda x.green(x)$ until we know of a specific object that is denoted by that expression. In work on epistemology [11] and AI approaches to planning under uncertainty [14, 15] the concept of a *rigid designator* plays an important role: it is a special name for an object (person, etc.) that always means the same thing, independent of context, and which can be used to specify a concrete operation. In our formulation, internal anchors are rigid designators that can serve as arguments for operations, but lambda expressions and existentially quantified variables are not. We introduce a fluent $KRD(A)$, where A may be a variable or a constant; it has value $True$ if and only if A is a constant. A precondition for any operation on an object will be that the robot knows a rigid designator for it.

IV. Planning and Inference

Rather than attempt to formulate and solve a POMDP exactly, we follow the effective approximation strategy, known in some circles as *model predictive control* and others as *replanning* [23], in which we repeatedly:

- Make an approximately optimal plan to achieve a goal (specified in belief space) given the current belief state;
- Execute the first step of the plan; then
- Make an observation and use it to update the belief.

Rather than making a conditional plan, which requires branching both on actions and observations, we plan in a determinized model [16] in which it is assumed that observation actions result in the most likely observation. Because the goal is in belief space (typically, to believe with high probability that some desired world state holds) the plans will contain actions that gain information, and if those actions result in unexpected observations, a new plan will be made based on the new belief.

We assume a regression-based (backward) planning algorithm that uses STRIPS-like rules with preconditions and results in the form of belief fluents, and with variable values drawn from discrete and continuous domains. We augment the planner with inference rules that can be used during backward chaining and play the role of axioms, and assume the planner can operate hierarchically, postponing detailed planning until more information is available.

a) Basic mechanisms: Planning and inference rules have the form

precond: $(\psi_1(\theta) = u_1), \ldots, (\psi_l(\theta) = u_l)$
result: $(\phi_1(\theta) = v_1), \ldots, (\phi_k(\theta) = v_k)$

The θ arguments are vectors of variables; the ϕ_i and ψ_i are fluents that may have constants or elements of θ as arguments; the v_i, u_i are either constants or elements of θ. When an operator is applied during the search, some variables in θ are bound by matching the operator's results to fluents in the goal. There may be additional variables in θ that are not yet determined; these represent the variety of ways that the operation can be carried out to obtain the same result. Physical operations are accompanied by an executable procedure, parameterized by aspects of θ. We assign a cost to each action or inference step that is $-\log p$ where p is the probability that the action will have the desired outcome; by finding a plan that minimizes the sum of these costs, we will have found the open-loop plan that is most likely to achieve the goal.

When planning in belief space, it is typical for actions to have belief preconditions: for example, a robot cannot attempt to pick up an object unless its belief about the location of that object has low variance. However, when an object's pose is not yet well known, not only is it impossible to pick the object up, it is impossible to plan in detail for how to pick it up (which will depend on the object's pose, what other objects might surround it, etc.). For these reasons, we assume a hierarchical planning mechanism that is able to make plans at a high level of abstraction (by postponing preconditions in the style of Sacerdoti [17]), and begin refining the initial step and eventually taking primitive actions without planning in detail for later parts of the plan. This mechanism makes it possible to delay detailed planning for physical actions, such as picking up an object, until the high-level precondition of knowing a rigid designator has been achieved.

b) Inference for goal interpretation: Given these planning and reasoning mechanisms, the ability to reason about denoting expressions and to plan and execute actions in service of interpreting them can be implemented by defining a few new fluent types and inference rules.

If the robot has a subgoal of having a rigid designator for an object that it believes is denoted by some expression $expr$, one way to achieve it is by coming to believe that some particular object Obj has the relevant properties $Props$. This reasoning is described in the inference rule below:

precond: $B(\text{Holds}(Props, Obj), P_r)$,
$B(\text{Den}(expr, Obj), P_p)$
$\text{PropsFor}(expr) = Props$
result: $B(\text{Den}(expr, Obj), P_r)$
$\text{KRD}(Obj)$

The function PropsFor determines which object properties would be useful to know in order to determine the denotation of the expression; the cost of this inference rule (log probability of its success) depends on P_p, the prior probability that object Obj has properties $Props$.

An alternative strategy for achieving the same subgoal, which applies even when there is no object with a reasonable prior probability of satisfying the expression, is to search for

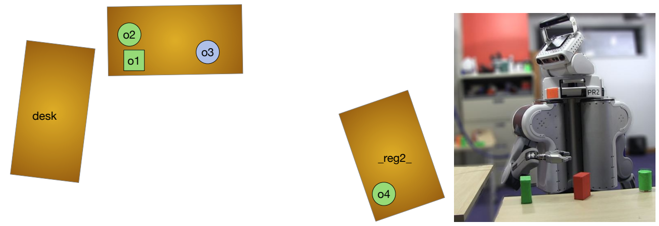

Fig. 2: Scenarios for illustrative example and robot experiment

such an object in regions of space that have not previously been explored:

precond: $\text{BContents}(Region, P_r)$,
$B(\text{ExistsInRegion}(expr, Region), P_p)$
result: $B(\text{Den}(expr, Obj), P_r)$
$\text{KRD}(Obj)$

This rule specifies that, for some region of space, if we come to know its contents, it may yield a belief about an object that satisfies the denotation; again the cost of the inference step is related to the log probability that there is such an object in the region; this cost-based reasoning encourages the planner to select regions for search in which an appropriate object is most likely to occur.

Finally, we provide inference rules, such as the one below, that connect symbolic properties, such as $Green$, with underlying object properties such as $Color$, which will allow further planning steps to determine that in order to gather information about the color of an object, it is necessary to look at it.

precond: $B(\text{Color}(Obj), \mu, \Sigma, \Delta, P_r)$
$\text{ColorParams}(Prop, \mu, \Sigma, \Delta)$
result: $B(\text{Holds}(Prop, Obj), P_r)$

V. Illustrative Example

We illustrate the close coupling of physical actions with goal interpretation in an extended example, presented in simplified form for clarity. Assume we have a mobile manipulation robot that can move its base and arms, pick and place objects, and look at them. Consider the arrangement of objects shown in Figure 2(left). The robot has already observed the objects on the table in front of it, but is unaware of other objects in its environment. The belief state is described in the table, where each row represents an object, and the columns provide a compact description of a detailed numerical distribution over each attribute of the object. The type distribution is described with the most likely value and its probability; the other distributions are specified with a mean value and approximate variance; and "prior" is intended to describe the initial belief, which is a high-variance Gaussian in this belief state.

id	type d	pose d	color d	weight d
_o1_	box, .92	(1, 1), low	green, low	prior
_o2_	can, .80	(2, 2), low	green, low	prior
_o3_	can, .87	(3, 3), low	blue, low	prior
desk	table, 1.0	(1, 1), low	brown, low	prior

The robot is given the goal
$\exists o. B(\text{Den}(expr, o), .9) \wedge B(\text{On}(o, desk), .9)$
where $expr \equiv \lambda x. Can(x) \wedge Green(x) \wedge Heavy(x)$. Assume

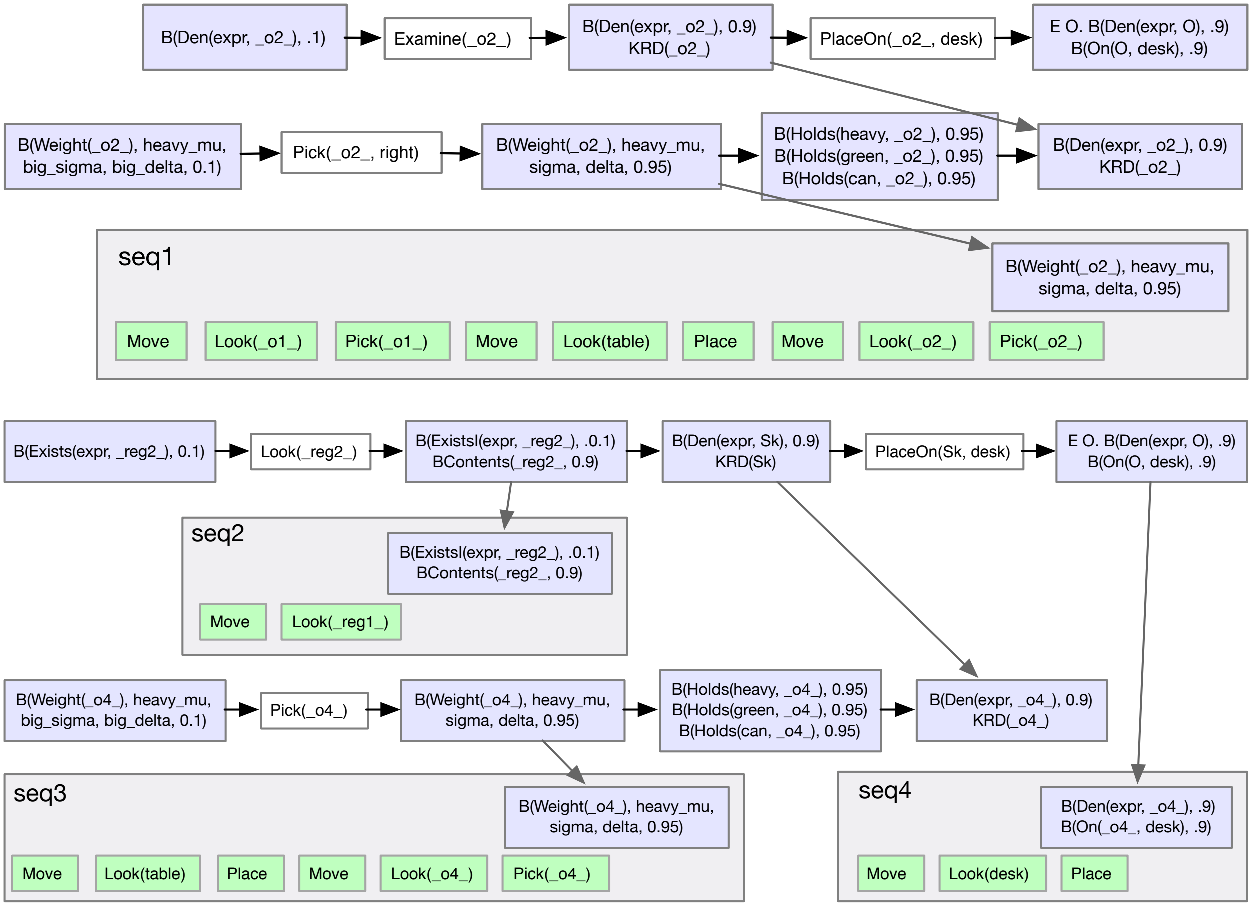

Fig. 3: Blue boxes contain belief formulas describing goals and subgoals. Each horizontal row of boxes represents a plan at some hierarchical level of the process. Clear boxes represent the execution of an abstract action; green boxes represent the execution of a primitive action; arrows between blue boxes represent inference steps.

that the denotation of *desk* is known to the robot and the definition of *Heavy* is an interval in weight space. Figure 3 illustrates the planning and reasoning process.

The highest-level plan has two abstract steps: examining an object with the internal anchor `_o2_`, and placing that same object on the desk. This plan was the most likely to succeed, which means that the object `_o2_` has a non-trivial probability of being a heavy green can. The hierarchical planning mechanism chooses the rightmost subgoal in that plan, and plans for it again, but using less abstract versions of the operators with more preconditions. The subgoal is $B(\mathrm{Den}(expr, \texttt{\_o2\_}), .9) \wedge \mathrm{KRD}(\texttt{\_o2\_})$. The planner determines, through several inference steps that it should pick up `_o2_`; this is because it already believes with high probability that it is green and a can, and so the weight is the crucial property to observe; then under the assumption of most likely observations, when it picks up the object, it will observe that it is heavy and satisfy the goal. Considerably more hierarchical planning, execution, and observation results in a sequence of primitive actions, summarized here by the sequence of green boxes, in which it moves the box, `_o1_`, out of the way so that it can finally move, look at `_o2_` to localize its pose, and then pick it up.

The robot gets an observation that `_o2_` weighs 100 g and updates its belief state accordingly (in fact, since it had to pick up `_o1_` in the process, it got a weight observation for it "for free," as well):

anchor	type d	pose d	color d	weight d
`_o1_`	box, .92	(10, 10), med	green, low	42, low
`_o2_`	can, .80	in hand	green, low	100, low
`_o3_`	can, .87	(3, 3), low	blue, low	prior
desk	table, 1.0	(1, 1), low	brown, low	prior

At this point, the pre-images of both plans on the stack are no longer true: the robot does not believe that `_o2_` has a significant probability of being heavy and therefore does not believe it could plausibly be denoted by *expr*.

The planner is re-invoked, resulting in the new abstract plan in the fourth row. This time, there are no objects that the robot knows about that could plausibly be the denotation of *expr*, so the plan is to look in some region of space (an index into a separate spatial data structure with internal anchor `_reg2_`) to find an object currently named by a constant *Sk*, and then to place that object on the desk. The rightmost subgoal is $B(\mathrm{ExistsIn}(expr, \texttt{\_reg2\_}), 0.1) \wedge \mathrm{BContents}(\texttt{\_reg2\_}, 0.9)$ which is to believe that an appropriate object could plausibly be in this region and to know its contents well. This goal is achieved through planning and execution of primitive actions (detailed reasoning is elided) until the robot makes an observation of a green object; the pose of this new object is sufficiently different from objects it already knows about it that the state estimator adds a new object, resulting in the following belief state:

anchor	type d	pose d	color d	weight d
_o1_	box, .92	(10, 10), med	green, low	42, low
_o2_	can, .80	in hand	green, low	100, low
_o3_	can, .87	(3, 3), low	blue, low	prior
desk	table, 1.0	(1, 1), low	brown, low	prior
_o4_	can, .91	(6, 6), low	green, low	prior

A line of reasoning similar to the one we saw before makes the object with anchor _o4_ most likely to be denoted by $expr$ and a plan is made to pick it up to observe its weight. After the sequence of actions labeled $seq4$ is executed (it has to execute a place action because it is still holding _o2_), the robot updates its belief about the weight of _o4_ to arrive at the following belief state

id	type d	pose d	color d	weight d
_o1_	box, .92	(10, 10), med	green, low	42, low
_o2_	can, .80	in hand	green, low	100, low
_o3_	can, .87	(3, 3), low	blue, low	prior
desk	table, 1.0	(1, 1), low	brown, low	prior
_o4_	can, .91	(6, 6), low	green, low	500, low

Finally, it places _o4_ on the desk, satisfying the goal.

VI. Robot implementation

We have integrated these mechanisms for reasoning about denotations with the pick-and-place capabilities of a PR2 robot for mobile manipulation, using the BHPN [9] planning and execution mechanism. The robot has a base, two arms and head, with total of 20 DOF. A Kinect sensor generates colored point clouds that are used for detecting objects; detections are categorized by type and are accompanied by a color observation, computed as the mean of the colors of the points associated to the object by the detector. The right wrist has a (very noisy) 6-axis force-torque sensor, which generates indirect observations of the weight of the object the robot is holding.

a) Simulation: Figure 4 illustrates the first scenario, with three objects on three tables, arranged so that at most one of the objects is in the field of view at a time. The goal is

$$\exists o.B(Den(\lambda x.Green(x), o), 0.9) \wedge B(In(o, table1), 0.9)$$

Table 1 is the table directly in front of the robot. The robot's initial belief includes the existence of the objects, but not their color. The robot plans to determine that the object $sodaE$ satisfies the denoting expression and to place it into the region. It observes the object, receives a color observation, and performs a belief update. The new belief (that $sodaE$ is probably red) means that the belief state is not in the pre-image of any of the plans on the stack. The robot replans and finds that object $sodaB$ is the most likely to satisfy the denoting expression, and so it moves and looks again, discovers that $sodaB$ is blue, and pops the plan stack once more. It tries once more, with object $sodaC$, discovers that it is green, then formulates and executes plans for picking up the object, moving, and placing it in the target region.

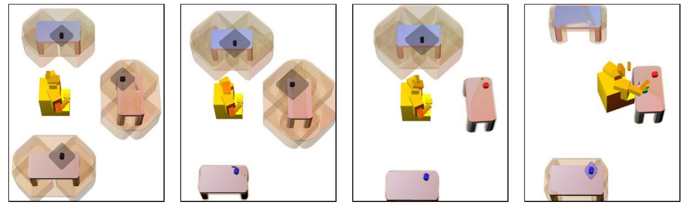

Fig. 4: Goal: a green object on right-hand table.

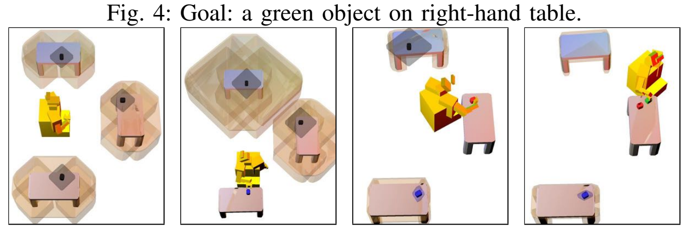

Fig. 5: Goal: a heavy object on right-hand table.

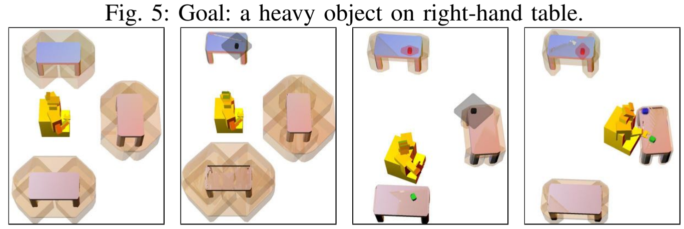

Fig. 6: Goal: a green object on right-hand table; no objects known in advance

In Figure 5, we begin with the same initial belief state, but the goal is now

$$\exists o.B(Den(\lambda x.Heavy(x) \wedge Soda(x), O), 0.9))$$
$$\wedge\, B(In(o, table1), 0.9)$$

This execution has a similar structure, in which the robot examines two objects before finding a third one satisfactory and putting it into the target region. However, now, because the objective is to find a heavy object, the robot must move over to each object in turn, and pick it up to weigh it. Once it has discovered a heavy object, the rest of the execution is actually simpler, because the robot finds that it is already holding the object it needs to place.

Finally, in Figure 6, we illustrate a situation in which the robot knows of the existence of tables in advance, and believes that objects are likely to be found on top of tables. Its goal is the same as in the first example:

$$\exists o.B(Den(\lambda x.Green(x), o), 0.9) \wedge B(In(o, table1), 0.9)\ .$$

The structure of the planning and execution that leads to the goal is almost exactly the same as for the first example, except that it uses the LookAtRegion operator to search in regions for objects; when it has thoroughly examined one region and not found any objects that satisfy the denoting expression, it pops the planning stack up as before.

b) Real robot: The same implementation runs on the physical robot. The first scenario corresponds to Figure 1, in which there are two oil bottles on a table, one mostly filled with oil, and one mostly empty. The initial poses of the bottles and tables were given with standard deviation $0.1m$. The robot was asked to achieve the goal

$$\exists o.B(Den(\lambda x.Heavy(x), o), 0.9) \wedge B(In(o, table1), 0.9) \ ,$$

which requires locating the heavy object and moving it to the table to its right. In this particular execution, the robot first looks at the oil bottle to its right to reduce pose uncertainty, then it picks it up to determining its weight. Based on the observation, the robot updates its belief about the weight of that object, and now believes, with high probability, that that bottle is not heavy. It replans, and decides to pick up the other bottle. It is nearly full, and therefore heavy enough to satisfy the specification. Thus, the robot places it on the table to the right.

We repeated the experiment multiple times, varying the poses of the oil bottles, including switching their left-to-right order. Due to noise in detections and non-determinism in the planner, we observed a variety of successful execution sequences with multiple look operations and different order of picking the bottles, which highlights the flexibility enabled by integrating the reference resolution with the physical planning. We also ran an experiment with the goal $\exists o.B(Den(\lambda x.Green(x), o), 0.9) \wedge B(In(o, table1), 0.9)$, using colored point clouds. The robot was given a green can, a green soda box, and a red box on a table in front of it, and had to move a green object to the right table (Figure 2). The robot reliably picks up the green soda box and moves it to the other table, even in the presence of rearrangement of the objects and of pose and type errors in the perception system, requiring different sequences of looking and motor operations. Videos of simulation and real robot experiments are available at `https://tinyurl.com/yc7na5es`.

c) Conclusions: It is critical to be able to communicate goals to robots in terms of object properties even when particular relevant objects are not known to the robot or the human. We have demonstrated that this capability can be achieved in a robust and flexible way through tight integration with state estimation and belief-state planning mechanisms that control the robot's physical and information-gathering actions.

d) Acknowledgments: We gratefully acknowledge support from NSF grants 1420316, 1523767 and 1723381, from AFOSR grant FA9550-17-1-0165, from Honda Research and Draper Laboratory. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our sponsors.

References

[1] Patrick Beeson, David Kortenkamp, R. Peter Bonasso, Andreas Persson, Amy Loutfi, and Jonathan P. Bona. An ontology-based symbol grounding system for human-robot interaction. In *Symposium on Artificial Intelligence for Human-Robot Interaction*, AAAI Fall Symposium Series, Washington, D.C., November 2014.

[2] Silvia Coradeschi and Alessandro Saffiotti. Perceptual anchoring of symbols for action. In *IJCAI*, 2001.

[3] Robin Deits, Stefanie Tellex, Pratiksha Thaker, Dimitar Simeonov, Thomas Kollar, and Nicholas Roy. Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction*, 2013.

[4] Manfred Eppe, Sean Trott, and Jerome Feldman. Exploiting deep semantics and compositionality of natural language for human-robot-interaction. In *Intelligent Robots and Systems (IROS)*, 2016.

[5] Peter Gorniak and Deb Roy. Probabilistic grounding of situated speech using plan recognition and reference resolution. *International Conference on Multimodal Interfaces*, 2005.

[6] Søren Hauberg, François Lauze, and Kim Steenstrup Pedersen. Unscented Kalman filtering on Riemannian manifolds. *Journal of Mathematical Imaging and Vision*, 46:103–120, 2012.

[7] Julian Hough and David Schlangen. It's not what you do, it's how you do it: Grounding uncertainty for a simple robot. In *HRI*, 2017.

[8] Saket Joshi, Paul Schermerhorn, Roni Khardon, and Matthias Scheutz. Abstract planning for reactive robots. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4379–4384, 05 2012.

[9] L. P. Kaelbling and T Lozano-Perez. Integrated task and motion planning in belief space. *International Journal of Robotics Research*, 2013.

[10] Huda Khayrallah, Sean Trott, and Jerome Feldman. Natural language for human robot interaction. In *International Conference on Human-Robot Interaction (HRI)*, 2015.

[11] Saul Kripke. *Naming and Necessity*. Blackwell Publishing, 1981.

[12] Séverin Lemaignan, Raquel Ros, Emrah Akin Sisbot, Rachid Alami, and Michael Beetz. Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *I. J. Social Robotics*, 4:181–199, 2012.

[13] Nikolaos Mavridis and Haiwei Dong. To ask or to sense? planning to integrate speech and sensorimotor acts. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*, 2012.

[14] Robert C. Moore. A formal theory of knowledge and action. In Robert C. Moore and Jerry R. Hobbs, editors, *Formal Theories of the Commonsense World*. Intellect Books, 1985.

[15] Leora Morgenstern. Knowledge preconditions for actions and plans. In *IJCAI*, 1987.

[16] Robert Platt, Russ Tedrake, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems*, 2010.

[17] Earl D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artif. Intell.*, 5:115–135, 1973.

[18] Niels Schuette, Brian Mac Namee, and John Kelleher. Robot perception errors and human resolution strategies in situated human-robot dialogue. *Advanced Robotics*, 2017.

[19] Kartik Talamadupula, J Benton, Subbarao Kambhampati, Paul W. Schermerhorn, and Matthias Scheutz. Planning for human-robot teaming in open worlds. *ACM TIST*, 1: 14, 11 2010.

[20] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 2011.

[21] Moritz Tenorth, Michael Beetz, Intelligent Autonomous Systems, and Technische Universität München. Towards practical and grounded knowledge representation systems for autonomous household robots. In *in Proceedings of the 1st International Workshop on Cognition for Technical Systems*, pages 6–8, 2008.

[22] Tom Williams and Matthias Scheutz. Resolution of referential ambiguity using Dempster-Shafer theoretic pragmatics. In *Proceedings of RSS*, 2017.

[23] Sung Wook Yoon, Alan Fern, and Robert Givan. Ff-replan: A baseline for probabilistic planning. In *ICAPS*, 2007.