Automatic Synthesis of Rules for Planning in Belief Space

Leslie Pack Kaelbling
MIT CSAIL
Cambridge, MA 02139
Email: lpk@mit.edu

I. Introduction

Robots operating in complex, unknown environments have to deal explicitly with uncertainty. Sensing is increasingly reliable, but remains inescapably local: robots cannot see, immediately, inside cupboards, under collapsed walls, or into nuclear containment vessels. Task planning in household and disaster-relief domains requires explicit consideration of uncertainty and the selection of actions at both the task and motion levels to support gathering information.

We have previously sketched an overall methodology for hierarchical task and motion planning in uncertain domains, and demonstrated it on a real robot [5]. A critical foundation of that methodology is the idea of planning in *belief space*: that is, the space of the robot's beliefs (represented as probability distributions and other data structures) about the state of its environment. For planning purposes, the initial state is a belief state and the goal is a set of belief states: for example, a goal might be for the robot to believe with probability greater than 0.99 that all of the groceries are put away in an acceptable location, or that there are no survivors remaining in the rubble.

Our approach is founded on the notion of integrating estimation, planning, and execution in a feedback loop. A plan is made, based on the current belief state; the first step is executed; an observation is obtained; the belief state is updated; the plan is recomputed, if necessary, etc. This approach allows planning to be approximate: it is important that the first step of the plan be useful, but the rest will be re-examined in light of the results of the first step. For this reason, we construct plans using a deterministic approximation of the probabilistic dynamics, which is discussed in detail in [5].

In order to support task planning in belief space, it had previously been necessary to write down a description of the domain by articulating the effects of different actions (moving an object, making an observation) in terms of their effects on the robot's belief state. Even experts in both the domain and in probabilistic reasoning found it difficult to write a correct and effective specification.

In this paper, we address that problem by providing a formalism for describing the stochastic transition and observation models that characterize the domain and an algorithm for converting those models into rules for planning in belief space. In addition to simplifying the problem of domain specification, this conversion process yields additional insight into conditions under which such a formalism could be correct and complete. This is a preliminary presentation of the work: the implementation and detailed solution is currently confined to discrete underlying state spaces.

A. Related work

There are two main classes of work closely related to symbolic planning in belief space.

First, is work on lifted, factored, and first-order partially observed Markov decision processes (POMDPs) [17, 14, 16]. Most of this work focuses on finding exact or near-optimal solutions in the form of complete *policies*. A policy maps all possible belief states to an action. The advantage of a policy is that it can be computed offline and used for fast look-up during execution. A disadvantage is that, in large domains, computing and even simply representing a reaction for every possible belief state can be wildly computationally intractable. By using lifted and factored representations, systematic regularities in the domain can be exploited to achieve some measure of compactness, but policy sizes still tend to grow very quickly.

Recently Srivastava et al. [15] have applied BLOG models to specifying POMDPs and have a thoughtful description of the semantics of observations, with a particular concern for the meanings of symbols in symbolic observations. In this work, we avoid those issues, by assuming observations do not contain quantifiers or designators.

Within the AI planning community, there has been a great deal of work on planning in a belief space corresponding to sets of underlying world states [3, 1]; here we concentrate on approaches that use explicit logical formulation of knowledge conditions. There is also a history of using logical representations to formalize knowledge preconditions for planning, starting informally with [8], then formalized in [10, 11]. This approach has been implemented in a planning system [13] and has been recently applied in a real robotic domain [4]. Recently, it has been demonstrated that planning under partial observability in non-probabilistic domains can be, under some assumptions, reduced to classical replanning, resulting in an approach that is very efficient, when it applies [2].

B. Simple example domain

As a running example, we consider a simple discrete domain that contains n *locations* numbered 0 through $n-1$, and m objects with names a, b, etc. Each object may be located in

any one of the locations. There are two primitive actions in this domain. The MOVE action takes two arguments, a source location and a target location. If there is an object at the source location, then it is moved, with probability 0.95 to the target location and with probability 0.05 to location 0. The LOOK action takes a single argument, which is a location. If there is an object at that location, then the robot receives an observation which is the name of that object, with probability 0.9, and observation *None* with probability 0.1. If there is no object there, then it observes the name of the object it is looking for[1] with probability 0.1, and *None* with probability 0.9.

Goals in this domain will be conditions on the belief state: conjunctions specifying that particular objects be believed to be in particular locations with some probability.

II. MODEL REPRESENTATION

We begin with a factored, lifted representation of the underlying POMDP. For simplicity in this paper, we assume that actions can either cause a transition in the world or generate an observation. Of course actions can have both types of effect, and the models can be generalized to account for that.

A. States and actions

Let

- R be a set of function symbols
- A be a set of action symbols
- I be a (possibly infinite) set of individuals (objects, locations, etc.)
- $\mathcal{R} = r(i_1, \ldots, i_j)$ for $r \in R, i_k \in I$, be a set of random variables, each of which has a finite domain of possible values
- The set of states $\mathcal{S}$ be the set of assignments of values to all the random variables in $\mathcal{R}$
- $\mathcal{A} = a(i_1, \ldots, i_j)$ for $a \in A, i_k \in I$, be a set of primitive actions.

The value of random variable $r(i_1, \ldots, i_j)$ may change over time; we will not always use an explicit time index, but will refer to these quantities as *random fluents* to emphasize the temporal dependence of their values. We will also make a pun and use $a(i_1, \ldots, i_j)$ in logical expressions to mean that the action named by the term was executed by the robot.

In our example domain, we have:

- $R = \{Loc, In, Clear\}$
- $A = \{Move, Look\}$
- $I = \{a, b, 0, 1, 2, 3, 4\}$

The semantics of $Loc(Obj) = v$ is that object Obj, which ranges over $\{a, b\}$, is in the location specified by v, which ranges over $\{0, \ldots, 4\}$.

The semantics of $In(Obj, Region) = v$ is that object Obj, which ranges over $\{a, b\}$, is in the one of the locations specified by $Region$, which ranges over $2^{\{0,\ldots,4\}}$ (that is, subsets of locations), if $v =$ **True** and is not in one of those locations if $v =$ **False**.

[1]This is a bit like the mirror of Erised.

The semantics of $Clear(Region) = v$ is that $Region$, which ranges over $2^{\{0,\ldots,4\}}$, contains no objects, if $v =$ **True** and contains at least one object if $v =$ **False**.

The action $\text{MOVE}(l_1, l_2)$ causes the robot to attempt to move the object at location l_1, if there is an object there, to location l_2. The action $\text{LOOK}(l_1)$ causes the robot to receive an observation of the object at location l_1.

B. Transition rules

To express the effects of actions on the state of the world, we use a rule formalism that is similar to the noisy indeterministic deictic (NID) rules of [12].

We assume that the transition dynamics of the domain are *factored*, so that $\Pr(S_{t+1} \mid S_t, A_t)$ can be written as:

$$\Pr(S_{t+1} = \langle v_1, \ldots, v_n \rangle \mid s_t, a_t) = \prod_{\rho_j \in \mathcal{R}} \Pr(\rho_j = v_j \mid s_t, a_t)$$

where $n = |\mathcal{R}|$, the number of possible random fluents.[2]

Furthermore, we assume that they are *lifted*, so that the form of the dependence is the same for all ρ that share a function symbol. This is in the style of models described by probabilistic relational models [7] and Blog [9] among others. So, for all $x_1, \ldots, x_i$, where the x variables range over elements of I,

$$\Pr(r_k(x_1, \ldots, x_i)_{t+1} \mid s_t, a_t) = \Pr(r_k(x_1, \ldots, x_i)_{t+1} \mid \psi(x_1, \ldots, x_j)_t, a_t) \ ,$$

where $\psi(x_1, \ldots, x_j) \subseteq \mathcal{R}$ is a set of random fluents whose arguments are individuals $x_1, \ldots, x_i$ and possibly some other related individuals, $x_{i+1}, \ldots, x_j$. Generally, we expect these dependencies to be relatively *sparse*, so that the value of a single random fluent at time $t+1$ depends on the values of a relatively small, constant number of random fluents at time t.

We assume that the distributions $\Pr(r_k(x_1, \ldots, x_i)_{t+1} \mid \psi(x_1, \ldots, x_j)_t, a_t)$ exhibit significant *context-specific independence*, and so are well described with a *rule-based* representation [6], with the distribution on $r_k(x_1, \ldots, x_i)_{t+1}$ being constant over some sets of assignments to its parents $\psi(x_1, \ldots, x_j))_t$. Thus, we express the transition model using a collection of rules of the form:

$$\forall x_1, \ldots, x_i. \big[\exists x_{i+1}, \ldots, x_j. \\ c_1(x_1, \ldots, x_j)_t = v_1 \wedge \cdots \wedge c_m(x_1, \ldots, x_j)_t = v_m \\ \wedge a(x_1, \ldots, x_j)_t\big] \rightarrow \\ \Pr(r(x_i, \ldots, x_i)_{t+1}) = \phi \ . \quad (1)$$

where the $c_\cdot \in R$, $a \in A$, and ϕ is a distribution over the domain of $r(x_1, \ldots, x_i)$.

A set of transition rules must have the property that, for any two rules with the same random fluent symbol r and

[2]If there are two or more fluents whose dynamics are not well modeled under this independence assumption, they can be grouped together into a single factor, with a dependent transition model.

action symbol a, the antecedent conditions must be mutually exclusive. The conditions need not be exhaustive, however: We make the STRIPS assumption that if there is no rule for describing how an aspect of the state changes, then it stays the same. So, if there is no rule for action a whose consequent is a random fluent with function symbol r and whose antecedent is satisfied in an initial state s_t, it is assumed that

$$\Pr(r(x_1, \ldots, x_i)_{t+1} = r(x_1, \ldots, x_i)_t \mid a(x_1, \ldots, x_j)_t) = 1 \ .$$

In general, there may be transitions that take place independent of the particular action being executed by the robot (e.g., objects being moved by other agents). An extension to this model, not addressed in this paper, would be to add another rule type for such exogenous events. A single action can affect multiple fluents. We can gather those effects together into a single action transition rule, by allowing multiple different random fluents in the effect: thus, we can characterize, either independently or jointly, the effects of the action on multiple fluents.

Here is a description of the MOVE operator, in a syntax that is easier to read, but whose semantics corresponds to the specification in equation 1.

MOVE$(obj, l_{start}, l_{target})$:
exists: $l_{start} \in \mathit{Locations} \setminus \{l_{target}\}$
pre: $\mathit{Loc}(obj) = l_{start}$
$\mathit{Clear}(sv(l_{start}, l_{target})) =$ **True**
effect: $\mathit{Loc}(obj) = \begin{cases} l_{target} & \text{w.p. } 1 - \mathit{moveErr} \\ 0 & \text{w.p. } \mathit{moveErr} \end{cases}$
prim: MOVEPRIMITIVE(l_{start}, l_{target})
cost: 1

This rule formalism is designed for use in backward chaining, in which case, the target location would be bound when the rule is applied, but the starting location remains free, which is why we have shown it as being existentially quantified. Here sv means "swept volume," and denotes the range of locations between the start and target locations, which should be free in order to move the object to the target location.

C. Observation rules

Actions of some types result in an observation that is conditionally dependent on the action and the state of the world. Let the set of possible observations be $\mathcal{O}$. In general, an observation $o \in \mathcal{O}$ might have a factored description that could make the observation model even more compact, but we do not assume that structure here.

The observation model has to specify $\Pr(O_t \mid s_t, a_t)$. We will specify it using rules, as with the transition model, but the rules have a slightly different form:

$$\forall x_1, \ldots, x_i. \big[\exists x_{i+1}, \ldots, x_j. \\ c_1(x_1, \ldots, x_j)_t = v_1 \wedge \cdots \wedge c_m(x_1, \ldots, x_j)_t = v_m \\ \wedge a(x_1, \ldots, x_j)_t\big] \rightarrow \\ \Pr(O_t = o \mid r(x_1, \ldots, x_i)_{t+1} = v) = f(x_1, \ldots, x_j, v, o) \ .$$

where f is a function mapping rule parameters $x_1, \ldots, x_j$, possible values v of $r(x_1, \ldots, x_i)_t$, and observations o into values in $[0, 1]$, subject to the constraint that, for all values of $x_1, \ldots, x_j$ and v,

$$\sum_o f(x_1, \ldots, x_j, v, o) = 1 \ .$$

This form is particularly useful when there are some physical conditions, characterized by the c random fluents, under which an observation can be made that is informative about random fluent r. For instance, the c conditions might be that the robot has an occlusion-free view of an object; and f might then characterize the noise in the observation model under the assumption that a good view is available.

Here is a description of the observation model for our exmaple domain. For simplicity we assume that each object has a string, $\mathit{name}(obj)$, associated with it, and that these strings can be observed.

To illustrate the role of conditions, we include one asserting that the lights are on, although it is not actually part of our example domain.

LOOK(obj, l_{look}):
pre: $\mathit{LightOn}() =$ **True**
result: $\Pr(O = o \mid \mathit{Loc}(obj) = v) =$
$\begin{cases} 0.9 & \text{if } o = \mathit{name}(obj) \text{ and } v = l_{look} \\ 0.1 & \text{if } o = \textbf{False} \text{ and } v = l_{look} \\ 0.05 & \text{if } o = \mathit{name}(obj) \text{ and } v \neq l_{look} \\ 0.95 & \text{if } o = \textbf{False} \text{ and } v \neq l_{look} \end{cases}$
prim: LOOKPRIMITIVE(l_{look})
cost: 1

This rule declares the existence of a LOOK action and says that, if the lights are on in the current state, and the actual location of the object in question is the same as the location the robot is looking at, then with high probability, the robot will get an observation that is the name of the object being looked at; otherwise, it will very likely get the observation **False**.

This is an approximation of the true model; it is useful for planning but would not be adequate for filtering. In fact, if object obj is not in the location we are looking at, then the observation will depend on what other objects might be in that location.

III. TRANSFORMATION TO BELIEF MODEL

Let the *belief state* at time t, b_t be a probability distribution over possible states in S. In a POMDP, whenever an action a is taken and an observation o made, the belief state is updated using the Bayesian filter equation:

$$b_{t+1}(s_j) \propto \Pr(o \mid a, s_j) \sum_i \Pr(s_j \mid s_i, a) b_t(s_i) \ .$$

The process characterizing the evolution of b is a Markov decision process (MDP) in the space of beliefs. This is the space in which we wish to plan, and so we provide a method

for translating the factored, lifted transition and observation models from the previous section into a factored, lifted representation of the belief space MDP. This representation is approximate: it allows for the construction of plans that are sound, in the sense that they have positive probability of reaching the goal, and correctly reports an upper bound on their cost; but it is neither complete nor optimal.

A. Belief fluents

Belief states are points in a what is generally a very high-dimensional space. Task planning methods are most appropriate in discretized spaces, but heedless discretization of belief space yields an intractably large discrete space. Here we present an approach that allows a goal-directed discretization to be constructed. We define a vocabulary of *belief fluents* for describing partitions of the belief space, and characterize the effects of transition and observation actions on the belief state, using this vocabulary of belief fluents. The vocabulary is designed to support means-ends reasoning via backward chaining.

A belief fluent has the form $B(\rho, v, \epsilon)$, where ρ is a random fluent, v is a value from the domain of ρ, and ϵ is a value in $[0, 1]$. It denotes the set of belief states in which the condition $\Pr(\rho = v) > 1 - \epsilon$ holds. We will not be able to completely characterize an actual belief state using belief fluents; this situation is analogous to the difficulty of completely characterizing geometric states using logical formulations. Goals and their pre-images (constructed during backward-chaining planning) will be conjunctions of belief fluents, which have the capacity of expressing a useful class of distinctions among belief states.

B. Transition rules

Given a rule describing the effects of an action on a class of random fluents, we can convert it to a set of rules characterizing the effect of that action on an associated belief fluent. Actions that cause domain transitions, but do not generate an observation, have deterministic effects on the belief state. We show the transformation by example on the following rule template:

TRANSRULE$(x_1, \ldots, x_j)$:
exists: $x_{i+1}, \ldots, x_j$
pre: $c_1(x_1, \ldots, x_j) = v_1, \ldots, c_m(x_1, \ldots, x_j) = v_m$
effect: $\Pr(r(x_1, \ldots, x_i) = u_1) = p_1$
$\ldots$
$\Pr(r(x_1, \ldots, x_i) = u_k) = p_k$
prim: PRIM$(x_1, \ldots, x_j)$
cost: c

If there are k possible values for the effect fluent, then we will construct k belief-space rules, one for each of the possible outcomes.

BTRANSRULE_L$(x_1, \ldots, x_j, \epsilon)$:
exists: $x_{i+1}, \ldots, x_j$
pre: $B(c_1(x_1, \ldots, x_j), v_1, \epsilon_1), \ldots,$
$B(c_m(x_1, \ldots, x_j), v_m, \epsilon_m)$
effect: $B(r(x_1, \ldots, x_i), u_l, \epsilon)$
prim: PRIM$(x_1, \ldots, x_j)$
cost: c

This rule says that, if the preconditions of the previous rule are true with probabilities lower-bounded by $1-\epsilon_1, \ldots, 1-\epsilon_k$, and the associated primitive action is executed, then the resulting fluent will have value u_l with probability $1 - \epsilon$. It remains only to characterize the relationship between these values. Assuming that the individual precondition random fluents are independent, we can show that

$$\Pr(r(x_1, \ldots, x_i)_{t+1} = v_l) = 1 - \epsilon \leq p_l \prod_{h=1}^{m} (1 - \epsilon_h)$$

We will use this rule for backward chaining, so if we desire to achieve a belief condition with probability at least $1 - \epsilon$, then this inequality provides a constraint on the values of the ϵ_h. In our current implementation, we set them all to be equal:

$$\epsilon_h = 1 - \left(\frac{1 - \epsilon}{p_l}\right)^{(1/m)}$$

Given some understanding of which belief conditions might be easier to achieve than others, it would be reasonable to do a non-uniform allocation of the uncertainty among the preconditions.

C. Observation rules

Similarly, we can take an observation rule and generate a belief-space operator from it. Recall that an observation rule has this form:

OBSRULE$(x_1, \ldots, x_j)$:
exists: $x_{i+1}, \ldots, x_j$
pre: $c_1(x_1, \ldots, x_j) = v_1, \ldots, c_m(x_1, \ldots, x_j) = v_m$
result: for all o_k and v_l
$\Pr(O = o_k \mid r(x_1, \ldots, x_i) = v_l) = f(x_1, \ldots, x_j, v_l, o_k)$
prim: PRIM$(x_1, \ldots, x_j)$
cost: c

This rule characterizes the likelihood of getting any observation o_k given that the target random fluent has value v_l. It can be used to construct rules that increase the degree of belief that the target fluent has value v_l. The effect of making an observation action on the belief state is stochastic: it depends on which particular observation will be received. Our approach is to construct a deterministic approximation of the stochastic domain, in which we are allowed to choose any of the desired results, but pay a cost of $-\log p$ for selecting an outcome with will occur with probability p.

We construct the following belief-space rule from the observation rule. It has three additional parameters: obs, v, and

ϵ. The values of v and ϵ will be bound during the backward-chaining planning search; the value of obs is a free variable, meaning that the operator can be instantiated for any possible value of the observation.

BOBSRULE$(x_1, \ldots, x_j, obs, v, \epsilon)$:
exists: $x_{i+1}, \ldots, x_j, obs$
pre: $B(r(x_1, \ldots, x_i), v, \epsilon_{prev})$
$B(c_1(x_1, \ldots, x_j), v_1, \epsilon_1), \ldots,$
$B(c_m(x_1, \ldots, x_j), v_m, \epsilon_m)$
effect: $B(r(x_1, \ldots, x_i), v, \epsilon)$
prim: PRIM$(x_1, \ldots, x_j)$
cost: $c - w \log p_{obs}$

The preconditions in this case are the conditions from the observation rule, as well as a previous condition on the belief that the target random fluent has the target value. Now, we need to find a relationship between the ϵ values in order to guarantee that the effect will hold. Using Bayes' rule and some conservative approximation, we find that the consequence will hold for

$$\epsilon_{prev} = \frac{\epsilon + p_c - \epsilon\ p_c(1 - p_{o|v,c})}{p_c(\epsilon\ p_{o|v,c} + (1 - \epsilon)p_{o|\neg v,c})},$$

where $p_c = (1 - \epsilon_h)^m$ is the probability that all the conditions hold, $p_{o|v,c} = f(x_1, \ldots, x_j, v, obs)$ is the observation probability from the model, and

$$p_{o|v,c} = \Pr(obs \mid r(x_1, \ldots, x_i) \neq v) \ .$$

This last quantity is difficult to compute; we approximate it, conservatively, with

$$p_{o|v,c} \approx \max_{u \neq v} \Pr(obs \mid r(x_1, \ldots, x_i) = u) \ .$$

The cost of this operation has two components: the original cost for executing the operation in the world, c, plus a cost that is related to the likelihood that the desired outcome will occur. A weighting factor w trades off cost of execution in the world against likelihood of plan success. The probability of getting the desired observation is the product of the probability that the preconditions are true with the probability that this particular observation will be obtained in that case:

$$p_{obs} = f(x_1, \ldots, x_j, v, obs)(1 - \epsilon_{prev})(1 - \epsilon_h)^m \ .$$

D. Entailment and contradiction

Because we are planning in very large or continuous domains, our symbolic planning formalism does not require add and delete lists to be enumerated in advance: in fact, the number of fluents that are potentially made true or false by an operation is possibly very large or unbounded. For this reason, we do limited inference inside the planning algorithm, based on pairwise entailment and contradiction relations among the fluents.

We are able to automatically derive entailment and contradiction relations between belief fluents from user-specified entailment and contradiction relations on the underlying random fluents. The relationships are these:

- For any two random fluents, if $\rho_1 = v_1$ entails $\rho_2 = v_2$, and $\epsilon_1 \leq \epsilon_2$ then $B(\rho_1, v_1, \epsilon_1)$ entails $B(\rho_2, v_2, \epsilon_2)$.
- For any two random fluents, if $\rho_1 = v_1$ contradicts $\rho_2 = v$, and $(1-\epsilon_1)+(1-\epsilon_2) \geq 1$ then $B(\rho_1, v_1, \epsilon_1)$ contradicts $B(\rho_2, v_2, \epsilon_2)$.

IV. EXAMPLE DOMAIN AND RESULTS

A. Remaining model specification

In order to have a set of transition rules that is moderately general for the underlying domain, we have to specify conditions under which values for *Clear* and *In* random fluents can be achieved. The rules are given below. They are *definitional* in the sense that there are no primitive operations associated with them; they reduce the *Clear* and *In* conditions to conjunctions of *Loc* fluents, which can be achieved with the MOVE operation.

MAKECLEAR$(region)$:
pre: $\forall\ obj.\ In(obj, region) =$ **False**
effect: $Clear(region) =$ **True** w.p. 1

MOVEOUT$(obj, region)$:
exists: $l \notin region$
pre: $Loc(obj) = l$
effect: $In(obj, region) =$ **False** w.p. 1

Additionally, we specify conditions under which pairs of *Loc*, *Clear*, and *In* fluents entail or contradict one another. This is straightforward.

B. Belief rules

The MOVE rule generates two belief rules (one for each outcome), and the MAKECLEAR, MOVEOUT and LOOK rules each generate a belief rule, which yields a total of 5 rules.

One additional feature of the belief rules is that they contain an applicability test. In transition rules, if $1 - \epsilon < p$, that is, the desired belief is greater than the accuracy of the transition being considered, then there is no degree of belief for the preconditions that will guarantee the desired result and the rule immediately detects its lack of applicability. Similarly, if ϵ_{prev} in an observation rule is not greater than ϵ, then making this observation is not increasing our certainty in the associated fluent, and so the operation need not be considered further.

C. Results

Consider a simple instance of our example domain in which there are 3 locations and two objects, *a* and *b*. In the initial belief state, the distribution on the location of *a* is $\{0 : 0.85, 1 : 0.05, 2 : 0.1\}$ and the distribution the location of *b* is $\{0 : 0..02, 1 : 0.96, 2 : 0.02\}$. The goal is $B(Loc(a), 1, 0.05)$; that is, to believe with probability at least 0.95 that object *a* is in location 1.

A partial search tree for this planning problem is shown in figure 1. Green nodes constitute the plan, found by backward-chaining A^* using the belief-space operators. Let us trace the plan, backward, from the root to the leaf. Each node describes a subset of belief space, in which all of the fluents in that

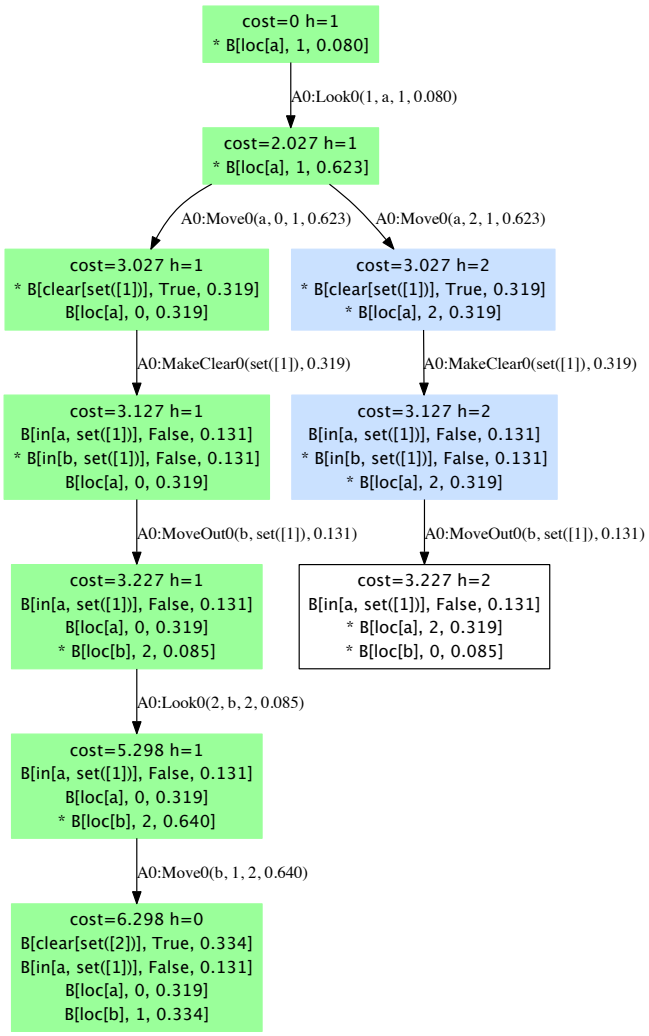

Fig. 1. Example search tree

node hold. The cost is the cost of the part of the plan that has been constructed so far, and the heuristic value is simply the number of fluents that are not true in the initial belief state. Fluents that are not true in the initial belief state are indicated with an asterisk.

The root node is the goal. Only a single operation is applicable to achieving the goal: to look in location 1, hoping to observe a. The last step cannot be a MOVE action, because the accuracy of moving is too low. But the regression of the goal under the LOOK action now only requires a to be in location 1 with a much weaker degree of belief, which makes a MOVE action applicable. It has a choice of moving the object from location 0 or from location 2, and will find that the cheaper plan moves it from location 0. However, in order for the MOVE operation to be sufficiently likely to succeed, we must believe that location 1 is clear, which, in turn, requires that both objects a and b be believed *not* to be in location 1. The condition is already satisfied for a, but not for b, so a subgoal of believing that b is in location 2 is established. This can be achieved most cost-effectively by putting in a LOOK to verify that b is in location 2, and then a MOVE to put it there. Read in the forward direction, with only primitive actions, the final plan is: $\text{MOVE}(1,2); \text{LOOK}(2); \text{MOVE}(0,1); \text{LOOK}(1)$.

If we increase the number of possible locations to 20, the search is still feasible. The immediate branching factor for MOVE operations is higher, but most of them are not useful and don't get pursued. In larger domains, it will be necessary to use some stronger heuristic guidance, which can generally come through the use of *generators*, which are procedures that order the choices of bindings for existential variables, so that we consider values that are heuristically likely to be useful earlier in the search.

V. REFLECTIONS AND FUTURE WORK

It will be critical to develop a formal characterization of the conditions under which this approach is correct and complete and to extend it to apply to a much broader class of problems. In this section, we briefly address these issues.

This approach is correct, in the sense that if a belief-space plan is found, then it has a non-zero chance of reaching the goal. The plans are not necessarily strictly optimal, even according to our notion of combined operation cost and negative log likelihood of success, because we may underestimate the success probability of some operations.

The approach is definitely not complete, in the sense of being able to achieve any goal in belief space: we cannot even articulate, for example, the goal of believing that locations 1 and 2 are nearly equally likely. An important question, though is whether the approach is complete with respect to the set of goals expressible as conjunctions of belief fluents. We conjecture that it is, but we have not yet shown it. Another important question is the behavior of the entire system, consisting of planning, execution, estimation, and replanning: it will be important to characterize conditions under which the system can be guaranteed to actually reach a given belief-space goal.

We believe it will be relatively straightforward to extend the approach to address continuous-valued random fluents with Gaussian and possibly other parametric distributions, as well as to handle factored and continuous observations.

Acknowledgement: This work was supported in part by the NSF under Grant No. 1117325. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. I also gratefully acknowledge support from ONR MURI grant N00014-09-1-1051, from AFOSR grant FA2386-10-1-4135 and from the Singapore Ministry of Education under a grant to the Singapore-MIT International Design Center.

REFERENCES

[1] P. Bertoli, A. Cimatti, M. Roveri, and P. Traverso. Planning in nondeterministic domains under partial observability via symbolic model

checking. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 473–478, 2001.

[2] Blai Bonet and Hector Geffner. Planning under partial observability by classical replanning: Theory and experiments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[3] Daniel Bryce, Subbarao Kambhampati, and David E. Smith. Planning graph heuristics for belief space search. *Journal of Artificial Intelligence Research*, 26:35–99, 2006.

[4] Andre Gaschler, Ronald Petrick, Torsten Kröger, Alois Knoll, and Oussama Khatib. Robot task planning with contingencies for run-time sensing. In *ICRA Workshop On Combining Task and Motion Planning*, 2013.

[5] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *International Journal of Robotics Research*, 2013.

[6] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[7] Daphne Koller and Avi Pfeffer. Probabilistic frame-based systems. In *AAAI/IAAI*, pages 580–587, 1998.

[8] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[9] Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. BLOG: Probabilistic models with unknown objects. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2005.

[10] Robert C. Moore. A formal theory of knowledge and action. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal Theories of the Commonsense World*. Ablex Publishing Company, Norwood, New Jersey, 1985.

[11] Leora Morgenstern. Knowledge preconditions for actions and plans. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 867–874, 1987.

[12] Hanna Pasula, Luke S. Zettlemoyer, and Leslie Pack Kaelbling. Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research*, 29, 2007.

[13] R. P. A. Petrick and F. Bacchus. Extending the knowledge-based approach to planning with incomplete information and sensing. In *International Conference on Automated Planning and Scheduling*, 2004.

[14] Scott Sanner and Kristian Kersting. Symbolic dynamic programming for first-order POMDPs. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2010.

[15] S. Srivastava, S. Russell, and A. Pfeffer. First-order models for pomdps. In *UAI Workshop on Statistical Relational Artificial Intelligence (StarAI)*, 2012.

[16] Chenggang Wang and Roni Khardon. Relational partially observable MDPs. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2010.

[17] Zahra Zamani, Scott Sanner, Pascal Poupart, and Kristian Kersting. Symbolic dynamic programming for continuous state and observation pomdps. In *NIPS*, pages 1403–1411, 2012.