Backward-Forward Search for Manipulation Planning Completeness Argument

Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling*

We give an argument that HBF is probabilistically complete over the class of hybrid planning problems that have a finite representation and are furthermore backwards robustly feasible if HBF's COS samplers are probabilistically complete.

1 Hybrid Planning Problems

Definition 1. A *hybrid planning problem* (HPP) $\langle s_0, \Gamma, \mathcal{A} \rangle$ is specified by an initial state $s_0 \in \mathcal{S}$, a finite goal set of constraints Γ, and a set of actions $\mathcal{A}$.

Definition 2. A finite sequence of actions $(a_1, a_2, ..., a_m) \in \mathcal{A} \times \mathcal{A} \times ...$ is a *solution* to an HPP if and only if the corresponding sequence of states $(s_0, s_1, ..., s_m) \in \mathcal{S}^n$ starting from s_0 and recursively constructed using $s_i = a_i.\mathit{eff}(s_{i-1})$ satisfies $\forall i \in [m]$, $s_{i-1} \in a_i.\mathit{con}$ and $s_m \in \Gamma$.

An alternative but equivalent way of defining a solution to an HPP is to use a sequence of state-sets instead of states as the witness for the plan's validity.

Definition 3. A state-set $\psi \subseteq \mathcal{S}$ is a collection states defined as the intersection of a finite number of constraints $\psi.\mathit{con} = \{C_1, ..., C_K\}$.

State-sets and states have a dual connection. For example, s_0 is a state while Γ is a state-set. Additionally, action constraints are state-sets while effects are partially specified states. Thus, when applying actions forward from s_0, it is natural to reason over sequences of states, while when considering actions backward from Γ, one usually reasons about sequences of state-sets derived using pre-images. As suggested, computing the pre-image for an action and state-set can be thought of as the inverse of applying that action to a state. Specifically, the pre-image of a state-set ψ under action a is a state-set ψ' where $\forall s' \in \psi'$, $s' \in a.\mathit{con}$ and $a.\mathit{eff}(s') \in \psi$. The resulting state-set ψ' is defined by a set of constraints derived from the constraints of a and ψ.

Note that the constraints are not generally simply the union of a's constraints and ψ's constraints. Instead, some of ψ's constraints are usually modified by

*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA caelan@mit.edu, tlp@mit.edu, lpk@mit.edu

a's effects. Recall that when applying a to a state, a's effects $a.\mathit{eff}$ enforce the state to have $V_{I(E)} = E$ for $E \in a.\mathit{eff}$. It may be the case that some constraint $C \in \psi.\mathit{con}$ involves one or mores variable mentioned by $a.\mathit{eff}$. In which case, when taking the pre-image, the value of these variables will not affect whether C is satisfied because these values are overwritten by a. Moreover, the variables in C are fixed to have the value given by the corresponding effect from a. Let $C[a.\mathit{eff}]$ give a new constraint C' that fixes constraint values $V_{I(E)} = E$ if $I(E) \in I(C)$. The state-set ψ' resulting from the pre-image is thus defined by the intersection of the following constraints:

$$\textsc{pre-img}(a, \psi) = a.\mathit{con} \cup \{C[a.\mathit{eff}] \mid \forall C \in \psi.\mathit{con}\}. \tag{1}$$

Finally, notice that if a constraint has all of its values filled in and is satisfied, then it by default holds for any state. Conversely, if a constraint has all of its values filled in and is not satisfied, then it does not hold in any state implying $\psi' = \emptyset$. Thus, we arrive at the alternative definition for a solution to an HPP that state-sets as the witness for a plan.

Definition 4. A finite sequence of actions $(a_1, a_2, ..., a_m) \in \mathcal{A} \times \mathcal{A} \times ...$ is a *solution* to an HPP if and only if the corresponding sequence of state-sets $(\psi_0, \psi_1, ..., \psi_m) \subseteq \mathcal{S}^n$ starting from $\psi_0 = \Gamma$ and recursively constructed using $\psi_i = \textsc{pre-img}(a_i, \psi_{i-1})$ satisfies $s_0 \in \psi_m$.

This definition will be useful when defining backwards robust feasibility which gives conditions on a sequence of sets of state-sets rather than a sequence of sets of states. We now will restrict our attention to HPPs that can be expressed in a parametrizable form. Recall that the initial definition made no assumptions on $\mathcal{A}$, and surely many of these unrestricted problems could not even be written down. We continue to use action templates as a way of parameterizing actions.

Definition 5. An *action-set* A is a parameterized set of actions instances from an action template $\textsc{ActionTemplate}_A$ with parameters $\boldsymbol{\theta} = (\theta_1, ..., \theta_k)$ from a parameter space $\boldsymbol{\Theta}_A = \Theta_1 \times ... \times \Theta_k$ subject to a functional constraint $g_A(\boldsymbol{\theta}) = 0$ that defines the set of valid parameters.

The set of action instances expressed in set builder notation is then:

$$A = \{\textsc{ActionTemplate}_A(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \boldsymbol{\Theta}_A,\ g_A(\boldsymbol{\theta}) = 0\}. \tag{2}$$

Definition 6. An HPP $\langle s_0, \Gamma, \mathcal{A}\rangle$ has a *finite representation* if there exists a finite set of action-sets $\{A_1, A_2, ..., A_T\}$ such that:

$$\mathcal{A} = \bigcup_{t=1}^{T} A. \tag{3}$$

As a modeling point, note that an action template can be used in more than one action-set. This may particularly make sense if a lower dimensional subset of actions from an action template connects with another action-set. Without such a division, problems that require moving between the two action-sets will likely not meet the feasibility condition because volume is computed using a single measure. Slight changes to the representation can subtly affect whether the problem is robustly feasible.

2 Backwards Robust Feasibility

Next, we identify a subclass of the finitely representable HPPs that meets a robustness condition. Algorithms that utilize sampling, such as in motion planning, require that the samples comprise a non-negligible volume of the sample space for the algorithm to be effective. Similarly, our robustness condition will ensure that the problem admits a sufficiently large number of solutions such that a non-negligible volume of actions lead to a solution for a sequence of action selections.

Note that an algorithm generally must sequentially select actions based on previous choices for it to effectively solve many of the HPPs we consider. Suppose an algorithm independently sampled actions to solve a HPP involving a robot tasked to pick up an object. The parameters of each move action and the final pick action are dependent in the sense that they must involve the same sequence of robot configurations for the resulting plan to be valid. Surely, an algorithm that uniformly sampled from the set of all move actions would have zero probability of generating a solution much less even a valid sequence of actions. Instead, a more effective algorithm would inductively sample actions based on the parameters of previously selected actions. And in particular, it would sample from the set of actions that can either be performed before or after some particular already chosen action.

This naturally leaves algorithms with two directions of sampling sequential actions: forward or backward. Again, this is another instance of the HPP duality mentioned earlier. Because HBF only samples predecessors of actions, we will solely introduce the terminology for backward sequential action selection. We will now define the set of action instances from action-set A that achieve a constraint C such that $a.\mathit{eff} \in C$. Furthermore, it is useful to further separate this set into the actions that to do this and have the same set of constraints satisfied by some state s. Adding in a target state to move towards ensures that sampled actions make progress. In order for actions to satisfy a constraint and have some of their own constraints satisfied by a state, several parameters often are fixed to particular values, decreasing the dimensionality of the parameter space. We will define I as the set of indices corresponding to the action constraints that are targeted to be satisfied. I could be any subset of constraint indices, so $I \in 2^{|A.con|}$ (where $|A.con|$ gives the number of constraints for actions from action-set A). Thus, let the following be the set of achieving actions:

$$\text{ACHIEVERS}(A, C; s, I) = \{a \in A \mid a.\mathit{eff} \in C \wedge \forall i \in I, s \in a_{.}con_i\}. \qquad (4)$$

As a practical note, many indices will lead to $\textsc{achievers}(A, C; s, I) = \emptyset$, so we do not often need to reason about exponentially many sets of achievers.

Finally, in order to make statements about volume, we define measures relative to a particular sets of actions. Let $\mu(\alpha; \Lambda)$ be a measure on subsets α of the parameterized set of actions Λ that assigns finite, nonzero measure to the parameter space of Λ. We will apply these measures to the sets of achievers presented above where the underlying parameter space of actions in the set may vary in dimensionality depending on A, C, s, and I.

We now present backwards robust feasibility. Again, there is similar condition for forward robust feasibility although we will not define it because it does not affect HBF's completeness. It would, however, be essential for an algorithm that sampled actions in the forward direction such as a basic progression search. Additionally, backwards robust feasibility is not specific to HBF; although we will not prove it here, a regression search is also probabilistically complete over the class of backward robustly feasibly HPPs.

Definition 7. An HPP $\langle s_0, \Gamma, \mathcal{A} \rangle$ is *m-backward robustly feasible* if and only if it has a finite representation $\{A_1, ..., A_T\}$ and there exists a sequence of action-sets $(A_{t_1}, ..., A_{t_m})$ and a nonempty set of robust solutions $\Pi \subseteq A_{t_1} \times ... \times A_{t_m}$ such that:

$$\forall (a_1, ..., a_i, ..., a_j, ..., a_m) \in \Pi, i, j \in [0, m-1], j \in [1, m], i < j : \tag{5}$$

the values:

- $s_i = a_i.\mathit{eff}(...a_1.\mathit{eff}(s_0))$
- $\psi_j = \textsc{pre-img}(a_{j+1}, ...\textsc{pre-img}(a_m, \Gamma))$
- $\alpha_j = \{a_j \in A_{t_j} \mid (a_1, ..., a_i, ..., a_j, a_{j+1}, ...a_m) \in \Pi\}$

satisfy the following:

$$\exists C \in \psi_i.con \text{ and } I \in 2^{|A.con|} : \tag{6}$$

where:

$$\Lambda = \textsc{achievers}(A_{t_j}, C; s, I) \tag{7}$$

satisfies

$$\Lambda \neq \emptyset \text{ and } \mu(\Lambda \cap \alpha_j, \Lambda) > 0. \tag{8}$$

Intuitively, a HPP is backward robustly feasible if it has sufficiently many solutions, from the same sequence of action-sets, where any contiguous subsequence on a solution that begins at s_i and ends at ψ_j has a non-negligible volume of actions, in some achieving set for a constraint in ψ_j, that move one step from ψ_j towards s_i. The result of this property is that for backwards robustly feasible problems, there is sampling strategy that will make progress moving from a state-set towards a state for any pair on a subset of the solutions.

Note that, despite its nuanced structure, this is a relatively weak robustness condition because only makes assumptions about a family of solutions to the problem instead of the full state-space, and moreover it does not say anything about the volumes of actions on a plan moving forward. One could define a stronger condition that imply both robustness conditions and maybe that the HPP has additional structure; however, we are most interested in closely as possible characterizing the full set of HPPs over which HBF is probabilistically complete.

3 Probabilistic Completeness

Before proving that HBF is probabilistically complete over the class of backwards robustly feasible HPPs, we now need to discuss probabilistic completeness with respect to the sampling primitives contained in each COS.

The most generic COS samplers simply select a single action from ACHIEVERS. In which case, an action-set A is paired with a single COS sampler Ω that can identify a minimal dimensionality sample space for ACHIEVERS$(A, C; s, I)$. This sampler is probabilistically complete if for any subset of nonzero measure, it will produce an action sample in that subset with probability one in the limit as the number of samples goes to infinity. This can be easier be accomplished by a common random or deterministic sampling strategy that covers the space such as uniform rejection sampling.

These basic samplers, while probabilistically complete under a smaller number of assumptions, are not effective in the manipulation HPPs that we focused on solving in our experiments. In event where the dynamics of an action-set A are of a particular form, we can frequently define more practical samplers. In particular, if A contains actions that can be performed sequentially, we can use samplers that efficiently reason about sequences of actions. Likewise, these samplers are provably effective under conditions specific to the dynamics of A. In the context of manipulation, the MOVE, MOVEHOLDING, and PUSH action-sets form dynamics that comprise linear movements in a metric space that is a subspace of the full configuration space. We thus were able to use motion planners for their COS sampler's to quickly sample actions. In order to analyze the completeness, we need a robustness condition that is more specific to the dynamics of the problem. In future work, we will give an example of robustness and completeness conditions in a pure manipulation planning domain. But for the proof in this paper, we will assume the COS samplers are the generic ones introduced above. We now return to discuss probabilistic completeness for hybrid planning algorithms.

Definition 8. A hybrid planning algorithm is *probabilistically complete* over a class of problems if and only if for every problem in the class, the probability of the algorithm finding a solution is one in the limit as the number of steps approaches infinity.

Before, giving the proof that HBF is probabilistically complete, we consider the these three simple lemmas meant to expose three main proof ideas.

Lemma 1. *For every m-backwards robustly feasible HPP $\langle s_0, \Gamma, \mathcal{A}\rangle$ with set of robust solutions Π where $m > 0$, $\forall (a_1, ..., a_m) \in \Pi$, the HPP $\langle a_1.\mathit{eff}(s_0), \Gamma, \mathcal{A}\rangle$ is $(m-1)$-backwards robustly feasible.*

Proof. For each solution $(a_1, ..., a_m) \in \Pi$, the set of robust solutions for the HPP $\langle a_1.\mathit{eff}(s_0), \Gamma, \mathcal{A}\rangle$ is $\Pi' = \{(a'_2, ..., a'_m) \mid (a_1, a'_2, ..., a'_m) \in \Pi\}$. This is because each $(a'_2, ..., a'_m) \in \Pi'$ is a contiguous subsequence of some solution in Π. Thus, the HPP $\langle a_1.\mathit{eff}(s_0), \Gamma, \mathcal{A}\rangle$ is $(m-1)$-backwards robustly feasible. □

Lemma 2. *Let $X_1, ..., X_m$ be a sequence of m independent, nonnegative random variables where $\lim_{t\to\infty} \Pr[X_i \leq t] = 1$ where t is the number of trials. For $X = \sum_{i=1}^{m} X_i$,*

$$\lim_{t\to\infty} \Pr[X \leq t] = 1. \tag{9}$$

Proof. We produce a simple lower bound by considering the event where each X_i succeeds in at most t/m trials.

$$\Pr[X \leq t] \geq \Pi_{i=1}^{m} \Pr[X_i \leq t/m] \tag{10}$$

Using multiplicity of limits and by a change of variables,

$$\lim_{t\to\infty} \Pr[X \leq t] \geq \Pi_{i=1}^{m} \lim_{k\to\infty} \Pr[X_i \leq k] \tag{11}$$

$$= 1 \tag{12}$$

□

Lemma 3. *Suppose a finite, but otherwise arbitrary, the number of trials t performed in steps n is given by $t(n)$. Furthermore, suppose $t(n)$ is such that a finite amount, but otherwise arbitrary, number of steps are performed before each trial. Then, for any function f, $\lim_{n\to\infty} f(t(n)) = \lim_{t\to\infty} f(t)$.*

Proof. Because trials and steps are both nonnegative integers, and a finite number of steps happen before each trial, $\lim_{n\to\infty} t(n) = \infty$. By composition of limits, $\lim_{n\to\infty} f(t(n)) = \lim_{t\to\infty} f(t)$. □

We will now prove that HBF is probabilistically complete over the class of backward robustly feasible HPPs. The argument is composed of two theorems. The first shows that the reachability graph grown from the start state of a backwards robustly feasible HPP will eventually contain a successor of the start that itself is a smaller backwards robustly feasible HPP. The second theorem inductively applies the first theorem to show that through repeated constructions of the reachability graph, HBF will find a solution with high probability.

Theorem 1. *For any m-backward robustly feasible hybrid planning problem $\langle s_0, \Gamma, \mathcal{A}\rangle$ and $\forall \tau > 0$, the reachability graph $G_0 = \textsc{ReachabilityGraph}(s_0, \Gamma)$ will contain an edge for an action a_1 such that a_1 is applicable in s_0 and the hybrid planning problem $\langle a_1.\mathit{eff}(s_0), \Gamma, \mathcal{A}\rangle$ is $(m-1)$-backward robustly feasible* *with probability one as the number of calls to* GROW-RG(G_0, Σ, τ) *approaches infinity if its* COS *samplers are probabilistically complete.*

Proof. Because the problem is m-backward robustly feasible, there exists a set of solutions that can collectively be characterized as sequentially achieving m sets of constraints $\mathcal{X}_1, ..., \mathcal{X}_m$ that are necessary to perform a future action or satisfy the goal constraints. We will assume that m is minimal, so on each length m robust solution, there does not exists an unnecessary action a_i, that fails to achieve any C in the adjacent pre-image PRE-IMG$(a_{i+1}, ...\text{PRE-IMG}(a_m, \Gamma))$. Thus, the first action a on each robust solution achieves a $C_1 \in \mathcal{X}_1$. But this C_1 is only necessary itself because either it is a constraint for some action that is on a solution achieving some later $C_i \in \mathcal{X}_i$ or $C_1 \in \Gamma$. By inductively tracing this sequence of constraints, we can identify a subsequence of sets of constraints $\mathcal{X}_1, \mathcal{X}_{i_2}, ..., \mathcal{X}_{i_k}$ that always starts from $\mathcal{X}_1$ and ends at some $\mathcal{X}_{i_k} \subseteq \Gamma$. This is usually a proper subsequence because actions and the goal may have several constraints that will also need to be satisfied while the subsequence just identifies a single constraint for each action or the goal.

We show that though its backwards search through constraint space, the GROW-RG(G_0, Σ, ∞) procedure will almost surely be able to identify a subsequence $(C_1, C_{i_2}, ..., C_{i_k}) \in \mathcal{X}_1 \times \mathcal{X}_{i_2} \times ... \times \mathcal{X}_{i_k}$. The procedure's queue is initialized with each goal constraint implying it will contain $\mathcal{X}_{i_k}$. We can then inductively apply the following argument to C_{i_j} starting with C_{i_k}. Let the first $C_{i_j} \in$. For each C_{i_j}, $\exists A, I$ such that for Λ_j = ACHIEVERS$(A, C_{i_j}; s_0, I)$, $\alpha_j = \{a \in \Lambda_j \mid a.con \cap \mathcal{X}_{i_{j-1}} \neq \emptyset\}$, $\mu(\alpha_j; \Lambda_j) > 0$ because these constraints are from a robust set of solutions. Because each the COS samplers are probabilistically complete and α has nonzero volume, the probability that a drawn sample $a \in \Lambda$ is also $a \in \alpha$ is one as the number of samples taken goes to infinity.

Because this process is done sequentially, by lemma **??**, as the number of sampling attempts for $(C_1, C_{i_2}, ..., C_{i_k})$ goes to infinity, the probability that $a_1 \in \alpha_1$ has been sampled is one. Recall that process happens on a persistent queue where $O(1)$ action samples are drawn from C_{i_j} on each iteration it is processed. Afterwards, each other constraint on the queue will be processed before returning to C_{i_j}. Although the queue may increase in size between times C_{i_j} is processed, only a finite number of other constraints are processed. Thus by lemma 3, as the number of iterations also goes to infinity, the probability of sampling a_1 is one. Moreover, for $\tau > 0$, as the number of calls to GROW-RG(G_0, Σ, ∞) goes to infinity, the probability of sampling a_1 is one, because each call performs at least one iteration. Finally, note by lemma 1, the HPP $\langle a_1.\mathit{eff}(s_0), \Gamma, \mathcal{A}\rangle$ is $(m-1)$-backwards robustly feasible. □

Theorem 2. HBF *is probabilistically complete over the class of backward robustly feasible hybrid planning problems if its* COS *samplers are themselves probabilistically complete.*

Proof. Consider any m-backward robustly feasible hybrid planning problem $\langle s_0, \Gamma, \mathcal{A}\rangle$. Recall that the top-level search of HBF is a hill-climbing procedure which, upon discovery of a state s' with a heuristic value h' lower than previous minimum h_{min}, conducts a simultaneous persistent search from both s' and s_0. The heuristic function $\text{H}(s, \Gamma)$ always returns a nonnegative integer or infinity as an estimate of the minimum number of actions from s to reach a state satisfying the goal constraints Γ. Thus h_{min} is a monotonically decreasing sequence of integers lower bounded by zero. This implies that the sequence is always finite. Furthermore, the number of steps taken before reaching the end of the sequence is finite.

Consider when the final heuristic value in this sequence not from a goal state is achieved. Without loss of generality, assume that no search nodes descending from s' will satisfy Γ, and therefore s' is a dead end. We now focus on the search starting from s_0 which will proceed interrupted because this search begins at the smallest h_{min} not from a goal state. This search will find a plan if it can discover a sequence of m states such that each $s_i = a_i.\mathit{eff}(s_{i-1})$ for $a_i \in \mathcal{A}$ and the resulting hybrid planning problem $\langle s_i, \Gamma, \mathcal{A} \rangle$ is $(m-i)$-backward robustly feasible. The search terminates when it reaches a 0-backward robustly feasible HPP $\langle s_m, \Gamma, \mathcal{A} \rangle$ which implies $s_m \in \Gamma$.

Inductively applying theorem 1, $G_{i-1} = \text{REACHABILITYGRAPH}(s_i, \Gamma)$ will contain an action a_i such that $s_i = a_i.\mathit{eff}(s_{i-1})$ with probability one as the number of calls to GROW-RG(G_{i-1}, Σ, τ) goes to infinity. The number of calls (i.e. trials) the search collectively makes before success is the sum of the number of calls to each GROW-RG(G_i, Σ, τ) for $i \in [0, m-1]$. By lemma 2, as the number of calls, goes to infinity, the probability that G_{m-1} generated s_m, causing the search to be succeed, also goes to infinity. Before the next call to GROW-RG(G_{i-1}, Σ, τ), the search will process all the nodes that were added to the queue before the previous call. Although the number of nodes processed between calls may grow as new nodes are created and added to the queue, it will remain finite. Additionally, the number of steps to process each node is finite because GROW-RG(G_{i-1}, Σ, τ) takes $O(\tau)$ steps at a time and computing the heuristic runs in polynomial time of the size of the node's current reachability graph. By lemma 3, this probability that the search is successful as the number of steps goes to infinity is also one because a finite number of steps (including the steps taken before the last search restart) happen before each call to GROW-RG. Thus, HBF is probabilistically complete.

□