Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Hierarchical Expertise Level Modeling for User Specific Contrastive Explanations

Sarath Sreedharan, Siddharth Srivastava and **Subbarao Kambhampati**
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ 85281 USA
{ ssreedh3, siddharths, rao } @ asu.edu

Abstract

There is a growing interest within the AI research community in developing autonomous systems capable of explaining their behavior to users. However, the problem of computing explanations for users of different levels of expertise has received little research attention. We propose an approach for addressing this problem by representing the user's understanding of the task as an abstraction of the domain model that the planner uses. We present algorithms for generating minimal explanations in cases where this abstract human model is not known. We reduce the problem of generating an explanation to a search over the space of abstract models and show that while the complete problem is NP-hard, a greedy algorithm can provide good approximations of the optimal solution. We also empirically show that our approach can efficiently compute explanations for a variety of problems.

1 Introduction

AI systems have the potential to transform society by assisting humans in diverse situations ranging from extraplanetary exploration to assisted living. In order to achieve this potential, however, humans working with such systems need to be able to understand them just as they would understand human team members. This presents a number of challenges because most humans do not understand AI algorithms and their behavior at the same intuitive level that they understand other humans. Recently, there have been attempts to bridge this gap by developing systems capable of explaining their behavior. Most recently [Chakraborti *et al.*, 2017] formulated the problem of explaining plans as that of model reconciliation. Their approach relied on identifying ways of bringing the human model (i.e the explainee model) closer to the robot model so that the plan in question appears optimal in the new model. Their work looked at scenarios in which the human used a model of the domain that was at the same level of fidelity as the one used by the agent to generate the plan. This approach, unfortunately, did not capture scenarios where the human possessed a lower level of expertise and thus used a more "abstract" or coarser representation of the model as compared to the AI agent.

In this paper, we propose a new approach to this problem where the agent explains its ongoing or planned behavior to humans with differing levels of expertise. We consider explanations in the framework of counterfactual reasoning, where a user who is confused by the agent's activity (or proposed activity) presents alternative behavior that they would have expected the agent to execute. This aligns with the widely held belief that humans expect explanations to be *contrastive* [Miller, 2017]. In keeping with the terminology used in social sciences literature we will call the set of alternative behaviors as *foils* to the proposed robot behavior.

For instance, consider a mission-control operator who needs to supervise the activity of an autonomous robot on Mars in the midst of a sandstorm that could present valuable data for analysis. If the robot proposes to go back to the base before going to a vantage point for observing the storm, the operator would naturally be perplexed, and may be motivated to ask, why doesn't the robot go directly to the vantage point?! Similarly, a human team member at a manufacturing plant may be perplexed by a robot's unnecessary detours while assembling an automobile engine. Not only do such situations involve personnel with varying skill levels, they also place a premium on the size of explanations.

A natural interaction would have the robot present an explanation about why the human's counterfactual suggestion would not apply in the current situation. This explanation could involve facts about the environment as well as about the robot's constraints. E.g., "I need to get a new battery pack to observe the sandstorm for at least 30 minutes without interruption". Such explanations need to be attuned to the level of understanding of the human involved. If the operator happens to be the lead designer of the robot's sequential decision-making engine, the robot could provide more specific information, e.g. "I am carrying battery-pack #00920", because this operator knows that some battery packs wouldn't allow it to carry out the full observation.

In this paper we present the **Hierarchical Expertise-Level Modeling** or the **HELM** approach for facilitating such context and user-specific explanations. We assume that the human user's understanding of the task is an abstraction of the model used by the robot. HELM generates the appropriate explanation by searching through a *model lattice* of possible abstractions of the agent's model. Each model within this lattice represents a different level of understanding of the task,

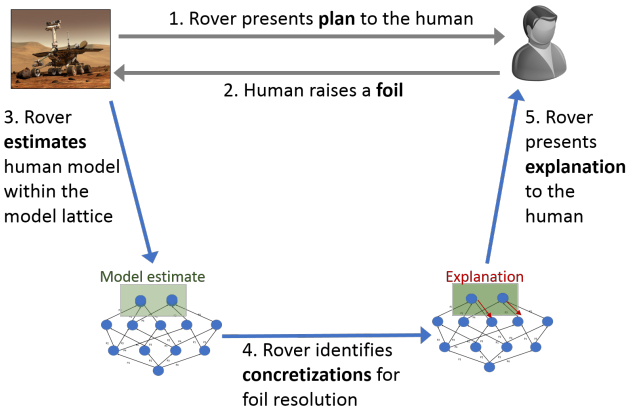

Figure 1: An illustration of the hierarchical explanation process. The human observer who views the task at a higher level of abstraction expects the rover to execute a different plan from the one chosen by the rover. The rover presents the human with an explanation it believes will help resolve the foils in the human's updated model.

with the highest fidelity representation (corresponding to the most detailed understanding of the domain used by the robot) forming the base of the lattice and the model representing the most naive understanding of the task (for example one held by a lay user) forming the highest node. Since the user's level of expertise is unknown to the agent, our algorithm estimates the human model before searching for an explanation. While we assume a closed form model in this paper, we plan to extend our approach to arbitrary generative models or simulations in future work.

Our explanations consist of information that may be absent in the user's abstract model, and show why the foil doesn't apply in the true situation. These explanations will cause the user's model to shift to a more accurate model in the lattice (and ultimately achieve model reconciliation). We will refer to model updates constituting these explanations as *model concretizations*. Our framework can also be extended to situations where a user's understanding is abstract and erroneous. In this paper, we focus on the fundamental aspects of the problem and restrict our attention to settings where the user's understanding is an abstraction of the actual situation.

Readers familiar with counter-example guided model checking (CEGAR) literature [Clarke *et al.*, 2000] or its applications in planning [Seipp and Helmert, 2013] will notice that our method of refining models is quite reminiscent of model refinement methods discussed in that literature. The foils we consider in our approach are equivalent to the counter-examples used by CEGAR methods and similar to these methods we too are looking for concretizations that refute these counter-examples. We provide a more detailed comparison between the methods in Section 5.

The rest of this paper is structured as follows. In Section 2, we present our formal framework. Section 3 covers different approaches for generating explanation and Section 4 presents empirical evaluation of these methods on standard IPC domains. In sections 5 and 6, we will discuss the related work and possible future directions.

2 Hierarchical Expertise-Level Models

In this work, we focus on abstractions that form models by projecting out state fluents. While the presentation in the following sections is equally valid for both predicate and propositional abstractions, we will focus on propositional abstractions to keep our formulation clear and concise. We will look at planning models of the form $\mathcal{M} = \langle P, S, A, I, G \rangle$ where P gives the set of state fluents, S the set of possible states, A the set of actions, I the initial state and G the goal. Each state $s \in S$ is uniquely represented by the set of propositions that are true in that state, i.e, $s \subseteq P$.

Each action $a \in A$ is associated with a set of preconditions prec_a that need to hold for the effects (e_a) of that action to be applied to a particular state. Each effect set e_a can be further separated into a set of add effects e_a^+ and a set of delete effects e_a^-. The result of executing an action a on a state s in this setting is defined as follows

$$a(S) = \begin{cases} (S \cup e_a^+) \setminus e_a^-, & \text{if } \text{prec}_a \subseteq S \\ S & \text{otherwise} \end{cases}$$

A plan π is defined as a sequence of actions ($\langle a_1, .., a_n \rangle$, n being the size of the plan), and a plan is said to solve $\mathcal{M}$ (i.e, $\pi(I) \models_{\mathcal{M}} G$) if $\pi(I) \supseteq G$.

Automated planning has a long tradition of employing abstraction both for plan generation (c.f [Sacerdoti, 1974]) and for generating heuristics (c.f [Seipp and Helmert, 2013; Keyder *et al.*, 2012]) and a number of different abstraction schemes have been proposed in these works. In fact, state abstractions as presented in this work have been widely used in pattern databases and are referred to as projections in that literature (c.f [Culberson and Schaeffer, 1998; Edelkamp, 2000]). Following works like [Seipp and Helmert, 2013; Backstrom and Jonsson, 2013], we will also use the concept of a transition system induced by the planning model to define state abstractions. Intuitively, a transition system constitutes a graph where the nodes represent possible states, and the edges capture the transitions between the states that are valid in the corresponding planning model. We refer the readers to the previously mentioned works for further analyses of state transition systems and their connection to abstractions.

Definition 1. *A set X is said to be a **propositional abstraction** of a set of states S with respect to some set of propositions Λ, if there exists a surjective mapping $f_\Lambda : S \to X$, such that for every state $s \in S$, there exists a state $f_\Lambda(s) \in X$ where $f_\Lambda(s) = s \setminus \Lambda$.*

For notational convenience we will refer to the set of states obtained by abstracting out the proposition set Λ from some set of states S as $[S]_{f_\Lambda}$.

Definition 2. *For a planning model $\mathcal{M} = \langle P, S, A, I, G \rangle$ with a corresponding transition system $\mathcal{T}$, a model $\mathcal{M}' = \langle P', S', A, I', G' \rangle$ with a transition system $\mathcal{T}'$ is considered an **abstraction of $\mathcal{M}$**, if there exists a set of propositions Λ, such that $P' = P - \Lambda$, $S' = [S]_{f_\Lambda}$, $I' = f_\Lambda(I)$, $G' = f_\Lambda(G)$ and for every transition $s_1 \xrightarrow{a} s_2$ in $\mathcal{T}$ corresponding to an action a, there exists an equivalent transition $(s_1 \setminus \Lambda) \xrightarrow{[a]_{f_\Lambda}} (s_2 \setminus \Lambda)$ in $\mathcal{T}'$, where $[a]_{f_\Lambda}$ is part*

of the new action set A'.

As per Definition 2, the abstract model is *complete* in the sense that all plans that were valid in the original model will have an equivalent plan in this new model. We will use the operator $\sqsubset$ to capture the fact that a model $\mathcal{M}$ is less abstract than the model $\mathcal{M}'$, i.e if $\mathcal{M} \sqsubset \mathcal{M}'$ then there exist a set of propositions Λ such that $\mathcal{M} = [\mathcal{M}']_{f_\Lambda}$. With the definition of abstraction and related notations in place, we are now ready to define a model lattice. We will use this lattice to both estimate the human model and to identify explanations.

Definition 3. *For a model $\mathcal{M}^{\#}$, the **model lattice $\mathcal{L}$** is a tuple of the form $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$, where $\mathbb{M}$ is the set of lattice nodes such that $\mathcal{M}^{\#} \in \mathbb{M}$ and $\forall\, \mathcal{M}' \in \mathbb{M}, \mathcal{M}^{\#} \sqsubseteq \mathcal{M}'$, $\mathbb{E}$ is the lattice edges, $\mathbb{P}$ is the superset of propositions considered for abstraction within this lattice and ℓ is a function mapping edges to labels. Additionally, for each edge $e_i = (\mathcal{M}_i, \mathcal{M}_j)$ there exists a proposition $p \in \mathbb{P}$ such that $[\mathcal{M}_i]_{f_p} = \mathcal{M}_j$ and $\ell(\mathcal{M}_i, \mathcal{M}_j) = p$.*

Thus each edge in this lattice corresponds to an abstraction formed by projecting out a single proposition (represented by the label of the edge). We can also define a concretization function γ_p that retrieves the model that was used to generate the given abstract model by projecting out the proposition p, i.e, $\gamma_p(\mathcal{M}) = \mathcal{M}'$ if $(\mathcal{M}', \mathcal{M}) \in \mathbb{E}$ and $\ell(\mathcal{M}', \mathcal{M}) = p$ else $\gamma_p(\mathcal{M}) = \mathcal{M}$.

Throughout the rest of this work, we will make some assumptions on the structure of the lattice $\mathcal{L}$ and the abstraction methods used by $\mathcal{L}$ to simplify our discussions. In this paper, we will focus on lattices where each node in $\mathbb{M}$ has an incoming edge for every proposition missing from its corresponding model. We will refer to lattices that satisfy this property as **Proposition Conserving** lattices. Additionally, we will call a proposition conserving lattice that contains an abstract node corresponding to each possible subset of $\mathbb{P}$ as the **Complete Lattice** for $\mathcal{M}$ given $\mathbb{P}$.

Formally, a lattice $\mathcal{L}$ is *proposition conserving*, iff for any model $\mathcal{M} \in \mathbb{M}$ and $\forall p \in \mathbb{P}$, if p is not in $P_{\mathcal{M}}$ then there exists a model $\mathcal{M}' \in \mathbb{M}$, such that $(\mathcal{M}', \mathcal{M}) \in \mathbb{E}$ and $\ell(\mathcal{M}', \mathcal{M}) = p$). Notice that enforcing conservation of propositions doesn't require any further assumptions about the human model and can be easily ensured by the agent generating the lattice.

We also assume that all abstraction functions used in generating the models in the lattice are commutative and idempotent, i.e., $[[\mathcal{M}]_{f_{p_1}}]_{f_{p_2}} = [[\mathcal{M}]_{f_{p_2}}]_{f_{p_1}}$ and $[[\mathcal{M}]_{f_{p_1}}]_{f_{p_1}} = [\mathcal{M}]_{f_{p_1}}$. Readers can refer to [Srivastava *et al.*, 2016] for a comprehensive list of ways to generate abstract models that satisfy these properties.

As mentioned earlier, we consider an explanation generation setting where the human observer uses a task model (denoted as $\mathcal{M}_H = \langle P_H, S_H, A_H, I_H, G_H \rangle$), that is a more abstract version of the robot's model ($\mathcal{M}_R = \langle P_H, S_R, A_R, I_R, G_R \rangle$). While the robot may not know $\mathcal{M}_H$, it knows that $\mathcal{M}_H$ is a member of the set $\mathbb{M}$ for the lattice $\mathcal{L}$. The human comes up with a **foil set F** $= \{\pi_1, \pi_2, ..., \pi_m\}$ that the robot needs to refute by providing an explanation E regarding the task. The explanation should contain information about specific domain properties (i.e., state fluents) that are missing from the human's model and how these properties affect different actions (For example, which actions use these propositions as preconditions and which ones generate/delete them). To illustrate the utility of such explanations consider an example involving a simplified version of the rover domain mentioned earlier.

Example 1. Let us suppose that the rover uses a modified version of the IPC rover domain [International Planning Competition, 2011] that also takes into account the battery level of the rover. Each rover operation has a different energy requirement, and the battery level needs to be above a predefined threshold for it to execute them, e.g., it can perform rock sampling only if the battery level is above 75%. Furthermore, the rover needs to visit the base station (i.e., the lander) and perform a reset action to recharge its batteries.

The rover knows that the human observer is at most ignorant of its energy requirements and/or storage capabilities. So the model lattice $\mathcal{L}$ needs to consider abstractions corresponding to the following propositions $\mathbb{P}$={battery_level_above_25_perc, battery_level_above_50_perc, battery_level_above_75_perc, full_store}. Figure 2 shows the lattice that the robot would use in this setting. Here we will create each abstract model by dropping a proposition from the more concrete model and by making the effects of action non-deterministic if the dropped predicate appears in the precondition. For example, if the action drop_store1 has effects of the form

$$\{\text{full\_store1},\ \text{store\_of\_store1}\} \rightarrow \{\neg\text{full\_store1},\ \text{empty\_store1}\}$$

Now in an abstract version of this model, if the proposition full_store1 is dropped the effect becomes

$$\{\ \text{store\_of\_store1}\} \rightarrow\ ND\{\ \text{empty\_store1}\}$$

Which now says that the action's effects are non-deterministic and executing drop_store1 may or may not turn the fluent empty_store1 true.

Here the robot presents the plan

$$\pi_R = \langle\ \text{navigate\_w0\_lander, reset\_at\_lander,}\ \text{navigate\_lander\_w1, sample\_rock\_store0\_w1}\rangle$$

and the observer responds by proposing the foil set

$$F = \{\langle\ \text{navigate\_w0\_w1, navigate\_lander\_w1,}\ \text{sample\_rock\_store0\_w1}\ \rangle\}$$

If the robot knew that the human was ignorant about all the battery level predicates and nothing else, the robot could help resolve the human confusion by informing them about the fact that action sample rock requires the battery to be above 75% (i.e describing the proposition battery_level_above_75_perc) and in this updated model the human foil can no longer achieve the goal. We can represent such an explanation using the set of propositions whose concretization is required to refute the given foils.

Definition 4. *An **explanation E** of size n for the human model $\mathcal{M}_H$ and a foil set F can be represented as a set of propositions of the form $E = \{p_1, ..., p_n\}$ such that*
$\forall \pi \in F, \pi(I_{\gamma_E(\mathcal{M}_H)}) \not\models_{\gamma_E(\mathcal{M}_H)} G_{\gamma_E(\mathcal{M}_H)}$
Where $\gamma_E(\mathcal{M}_H)$ is the model obtained by applying the concretizations corresponding to E on the model $\mathcal{M}_H$.

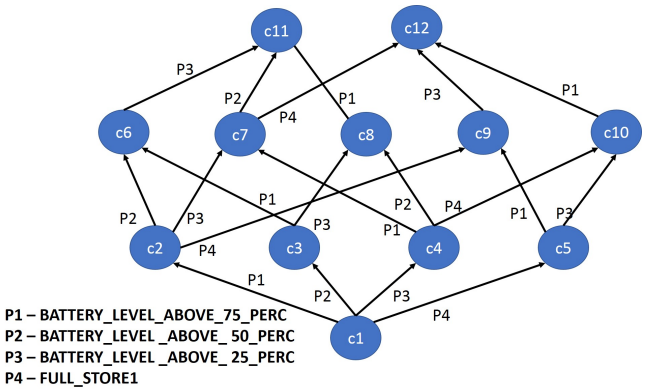

Figure 2: A possible abstraction lattice for the rover domain.

In Example 1, the rover would have difficulty coming up with a single explanation as it does not know $\mathcal{M}_H$. One possibility would be to restrict its attention to just the models that are consistent with the foils . In this scenario, this would correspond to $\{c6, c7, c9, c10, c11, c12\}$.

Now we need to find a way of generating explanations given this reduced set of models.

Proposition 1. Let $\mathcal{M}_i$ be some model in $\mathcal{L}$ such that $\mathcal{M}_H \sqsubseteq \mathcal{M}_i$. If E is a valid explanation for $\mathcal{M}_i$ and some foil set F, then E must also explain F for $\mathcal{M}_H$.

This proposition directly follows from the fact that for a proposition conserving lattice $\gamma_E(\mathcal{M}_i)$ will be a logical weaker model than $\gamma_E(\mathcal{M}_H)$. Next, we will define the concept of a minimal abstraction set for a given lattice $\mathcal{L}$ and foils F

Definition 5. *Given an the abstraction lattice* $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$ *the* **minimal abstraction set** $\mathbb{M}_{min}$ *is the supremum of all the models that are consistent with the foil set* F.
$\mathbb{M}_{min} = \sup\{\mathcal{M}_i | \mathcal{M}_i \in \mathbb{M}, \forall \pi \in F(\pi(I_{\mathcal{M}_i}) \models_{\mathcal{M}_i} G_{\mathcal{M}_i})\}$

In Example 1, the minimal abstract model set will be $\mathbb{M}_{min} = \{c11, c12\}$.

If we can find an explanation that is valid for all the models in $\mathbb{M}_{min}$ then by Proposition 1 it must work for $\mathbb{M}_H$ as well.

Proposition 2. For a given model lattice $\mathcal{L}$, the minimal abstraction set $\mathbb{M}_{min}$ and a set of foils F, there exists an explanation E such that $\forall\ \mathcal{M}' \in \mathbb{M}_{min}$ and $\forall \pi \in F$,
$\pi(I_{\gamma_E(\mathcal{M}')}) \not\models_{\gamma_E(\mathcal{M}')} G_{\gamma_E(\mathcal{M}')}$

It is easy to see why this property holds, as any explanation that involves concretizing all possible propositions in $\mathbb{P}$ satisfies this property.

In most cases, we would prefer to compute the least costly or the shortest explanation (if all concretizations are equally expensive) to the explainee. In the rover example, even if the human is unaware of multiple task details, the robot can easily resolve the explainee's doubts by just explaining the concretizations related to the proposition battery_level_above_75_perc without getting into other details. Describing the details of remaining propositions is unnecessary and in the worst case might leave the human feeling overwhelmed and confused. In this case, the explanation would just include information regarding battery levels and how to identify when the battery level is or above 75% and model updates like
sample_rock-has-precondition-battery_level_above_75_perc
sample_soil-has-precondition-battery_level_above_75_perc
...
Before delving into the optimization version of the problem, let us look at the complexity of the corresponding decision problem

Theorem 1. *Given a minimal abstraction set* $\mathbb{M}_{min}$*, a plan* π_R*, the set of propositions being abstracted* $\mathbb{P}$ *and the set of foils* F *for a model* $\mathcal{M}$*, the problem of identifying whether an explanation of size* k *exists for the complete lattice is* ***NP-complete****.*

Proof (Sketch). The fact that we can test the validity of the given explanation in polynomial time (size of the explanation is guaranteed to be smaller than $|\mathbb{P}|$) shows that the problem is in **NP**. We can show **NP-completeness** by reducing the set covering problem [Bernhard and Vygen, 2008] to an instance of the explanation generation problem. Let's consider a set covering problem with U as the universe set and S as the set of sub-collections. Now let us create an explanation generation problem where the set of foils F is equal to U and the propositions in the set $\mathbb{P}$ contain a proposition for each member of S. Additionally concretizing with respect to a proposition will resolve only the foils covered by its corresponding subset in S. For this setting, we can construct a fully connected proposition conserving lattice $\mathcal{L}$ of height $|S|$. Within the lattice, there exists a unique most abstract model where all the foils hold and a single most concrete model (where none of the foils hold). Now if we can come up with an explanation of size k in this setting, then this explanation corresponds to a set cover of size k. □

3 Generating Minimal Explanations

As mentioned earlier, we are interested in producing the minimal explanation. Additionally, in most domains, the cost of communicating the concretization details could vary among propositions. An explanation that involves a proposition that appears in every action definition might be harder to communicate than one that only uses a proposition that is part of the definition of a single action.

In addition to the actual size, the comprehensibility of the explanations may also depend on factors like human's mental load, the familiarity with the concepts captured by the propositions, etc.. To keep our discussions simple, we will restrict the cost of communicating an explanation to just the number of unique model updates this explanation would bring about in the human model.We will use the symbol C_p to represent the cost of communicating the changes related to the proposition p and also overload it to work on sets of propositions.

Now our problem is to find the cheapest explanation (represented as E_{min}) for a given set of foils F, and the minimal abstract model set $\mathbb{M}_{min}$. One possibility is to perform an A* search [Hart *et al.*, 1968] over the space of possible propositional concretizations to identify E_{min}. Each search state consists of the minimal set of abstract models for the human

model given the current explanation prefix. We will stop the search as soon as we find a state where the foils no longer hold for the current minimal set.

Proposition 3. Let $\mathbb{M}_{min}$ be the minimal abstraction set for a given lattice $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$ and foil set F. Then for a proposition p, the set $\widehat{\mathbb{M}}_{min}$ formed by applying the concretization corresponding to p on every element of $\mathbb{M}_{min}$ will be the minimal abstract set for $\widehat{\mathbb{M}}$ formed by applying the concretization γ_p on every element of $\mathbb{M}$ given F.

The above property implies that we don't need to look at the lattice $\mathcal{L}$ to recalculate minimal abstraction set after the application of every concretization function. We can also further simplify our problem by exploiting the fact that a particular propositional concretization resolves a foil (i.e., make the foil no longer valid) when it either adds a precondition (or a new condition for a conditional effect) or a goal fact that can not be satisfied by the foil. To concisely capture this idea we will introduce the concept of a foil resolution set to represent the subset of foils resolved by the concretization of a particular proposition.

Definition 6. *For a set of models $\mathbb{M}'$, a foil set F and a proposition p, the **resolution set** $\mathcal{R}_F(\mathbb{M}', p)$ gives the subset of foils that no longer holds in the concretized models, i.e* $\mathcal{R}_F(\mathbb{M}', p) = \{\pi | \pi \in F \wedge (\forall \mathcal{M}' \in \mathbb{M}'(\pi(I_{\gamma_p(\mathcal{M}')}) \not\models_{\gamma_p(\mathcal{M}')} G_{\gamma_p(\mathcal{M}')} \wedge \pi(I_{\mathcal{M}'}) \models_{\mathcal{M}'} G_{\mathcal{M}'}))\}$.

We will also use $\mathcal{R}_F$ to represent the set of foils resolved by a sequence of propositions

Proposition 4. For a set of model $\mathbb{M}'$ and a foil set F

$$\mathcal{R}_F(\mathbb{M}', \langle p_1, p_2 \rangle) = \mathcal{R}_F(\mathbb{M}', \langle p_1 \rangle) \cup \mathcal{R}_F(\mathbb{M}', \langle p_2 \rangle)$$

The above property implies that concretizing any n propositions cannot resolve foils that weren't resolved by the individual propositions. The idea of generating resolution sets are again closely related to the idea of resolving counter-examples and can be understood

Proposition 5. For two models $\mathcal{M}_1$, $\mathcal{M}_2$ and a set of foils F, if $\mathcal{M}_1 \sqsubseteq \mathcal{M}_2$ then for any proposition p, $\mathcal{R}_F(\{\mathcal{M}_1\}, p) \subseteq \mathcal{R}_F(\{\mathcal{M}_2\}, p)$

The above proposition ensures that if an explanation is the minimal one for $\mathbb{M}_{min}$, then it must be the minimal explanation for $\mathcal{M}_H$ as well.

These propositions will be instrumental in proving the effectiveness of our greedy algorithm described by Algorithm 1. In each iteration of this search, the algorithm greedily chooses the proposition that minimizes $\frac{C_p}{|F' \cap \mathcal{R}_F(\mathbb{M}', p)|}$, where F' is the set of unresolved foils at that iteration and the search ends when all foils are resolved.

Theorem 2. *The explanation $\widehat{E}$ generated by Algorithm 1 for a set of foils F and a lattice $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$ is less than or equal to $(\ln k) * C_{E_{min}}$, where $C_{E_{min}}$ is the cost of an optimal explanation and k represents the maximum number of foils that can be resolved by concretizing a single proposition, i.e, $k = \max_p |\mathcal{R}_F(\mathbb{M}_{min}, p)|$.*

Proof (Sketch). We will prove the above theorem by showing that Algorithm 1 corresponds to the greedy search algorithm for a weighted set cover problem. Consider a weighted set cover problem $\langle U, S, W \rangle$ such that the universe set $U = F$, the subcollections set S is defined as $S = \{s_p | p \in \mathbb{P}\}$ where $s_p = \mathcal{R}_F(\mathbb{M}_{min}, p)$ and the cost of each subset s_p is gives as $W(s_p) = C_p$. Proposition 4 ensures that the size of resolution set is a submodular and monotonic function. In this setting, the act of identifying a set of propositions that resolve the foil set is identical to coming up with a set cover for U in the new weighted set cover problem. Furthermore, we can show that the optimal set cover $\mathcal{C}_{opt}$ must correspond to the cheapest explanation E_{min} (We can prove this equivalence using Propositions 1,2 and 4, we are skipping the details of this proof due to space constraints). Algorithm 1 describes a greedy way of identifying the cheapest set cover for this weighted set cover problem and thus the minimal explanation for the original problem. For weighted set cover the above greedy algorithm is guaranteed to generate solutions that are at most $\ln k * W(\mathcal{C}_{opt})$ [Young, 2008], where $k = \max_{s \in S} |s|$ and this approximation guarantee will hold for E_{min} as well. □

Algorithm 1 Greedy Algorithm for Generating $\widehat{E}$

1: **procedure** GREEDY-EXP-SEARCH
2: *Input*: $\langle F, \mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle \rangle$
3: *Output*: Explanation $\widehat{E}$
4: *Procedure*:
5: curr_model = $\langle \mathbb{M}_{min}, F \rangle$
6: $\widehat{E} = \{\}$
7: $\mathbb{M}_{min}$ $\leftarrow$MinimalAbstractModels($\mathcal{L}, F$)
8: Precompute the resolution sets $\mathcal{R}_F(\mathbb{M}_{min}, p)$ for each $p \in \mathbb{P}$
9: **while** True **do**
10: $\mathbb{M}', F' =$ curr_model
11: **if** $|F'| = 0$ **then** return $\widehat{E}$ ▷ Return $\widehat{E}$ if all the foils are resolved
12: **else**
13: $p_{next} = \arg\min_p(\frac{C_p}{|F' \cap \mathcal{R}_F(\mathbb{M}', p)|})$
14: $\mathbb{M}_{new} = \{\gamma_{p_{next}}(\mathcal{M}) | \mathcal{M} \in \mathbb{M}'\}$
15: curr_model = $\langle \mathbb{M}_{new}, F \setminus \mathcal{R}_F(\mathbb{M}', p) \rangle$
16: $\widehat{E} = \widehat{E} \cup p$

We can use this algorithm to either generate solutions and or to calculate an inadmissible heuristic for the previously mentioned A* search. For the heuristic generation, we will further simplify the calculations (specifically step 8 in Algorithm 1) by considering an over-approximation of $\mathcal{R}_F$. Instead of considering the set of all foils resolved by concretizing each proposition p, we will consider the set of foils where p appears in the precondition of one of the actions in it. This set should be a superset for $\mathcal{R}_F$ for any proposition.

4 Empirical Evaluations

In our evaluation, we wanted to understand how effective our approaches were in terms of the conciseness of the explanations produced, the solution computation time and the useful-

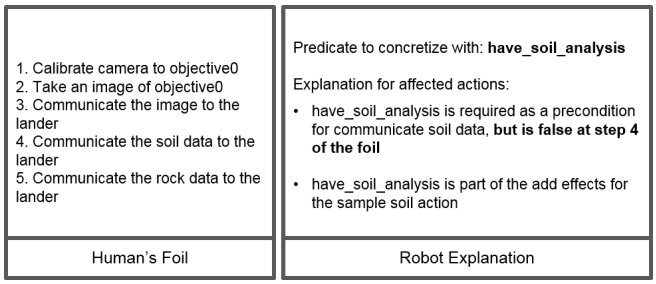

Figure 3: An example explanation generated by our system for IPC rover domain. The human incorrectly believes that the rover can communicate sample information without explicitly collecting any samples. While the abstraction lattice in this example was generated by projecting out upto 12 predicates, the search correctly identifies concretizations related to *(have_soil_analysis ?r - rover ?w - waypoint)* as the cheapest explanation ($C_E = 2$ as opposed to $C_{\mathbb{P}} = 55$)

ness of approximation. For the approximation, we were interested in identifying the trade-off between decrease in runtime vs. reduction in solution quality.

All three explanation methods discussed in this paper (blind, heuristic and greedy) were evaluated on five IPC benchmark domains[International Planning Competition, 2011]. All the experiments detailed in this section were run on an Ubuntu workstation with 64G RAM.

For each domain, we selected 30 problems from either available test sets or by using standard problem generators (the problems sizes were selected to reflect the size of previous IPC test problems). The lattice for each problem-domain pair was generated by randomly selecting 50% of domain predicates and then generating a fully connected proposition conserving lattice using that set of predicates. Since none of the models contained any conditional effects or negative preconditions, we created the abstract models by dropping the propositions to be abstracted from the domain models (which are complete for these domains). The foils were generated by selecting random models from the lattice and creating plans from these models that do not hold in the concrete model. Each search evaluated here, generates the set of proposition whose concretizations can resolve the foils set F. In actual applications, this set of propositions needs to be converted into an explanan (the actual message) by considering how this proposition is used in the robot model. Figure 3 shows the explanation generated by our approach for a problem in Rover domain.

The table in Figure 4 presents the results from our empirical evaluation on the IPC domains. The table shows the average cost/size of each explanation along with the time taken to generate them. Note that by size, we refer to the number of predicates that are part of the explanation while the cost reflects the total number of unique model updates induced by that explanation. We attempted explanation generation for foil set sizes of one, two and four per problem.

Our main conclusion is that heuristic search seems to outperform blind search in almost every problem and generates near-optimal solutions (Blind search always generates the minimal explanation). Further, we saw that greedy search outperformed heuristic search in most cases barring a few exceptions. The greedy search was able to make significant gains especially for higher foil set sizes. This is entirely expected due to the fact that step 8 in Algorithm 1 can be expensive for problems with long plans (but still polynomial). This expensive pre-computation pays off as we move to cases where E_{min} consists of multiple propositions. Additionally, we found out that greedy solutions were quite comparable to the optimal solutions with respect to their costs. For example in $|F| = 4$ for satellite domain, while the greedy solution cost took a penalty of $\sim 1.4\%$ the search time was reduced by $\sim 68\%$. Figure 5 plots the comparison between the time saved by the greedy search versus any loss in optimality incurred by the greedy search.

5 Related Work

There is increasing interest within the automated planning community to solve the problem of generating explanations for plans ([Fox *et al.*, 2017; Langley *et al.*, 2017]). Earlier works like [Seegebarth *et al.*, 2012; Bercher *et al.*, 2014; Kambhampati, 1990] looked at explanations as a way of describing the effects of plans, while works like [Sohrabi *et al.*, 2011; Meadows *et al.*, 2013] looked at plans itself as explanations for a set of observations. Another approach that has received a lot of interest recently is to view explanations as a way of achieving model reconciliation [Chakraborti *et al.*, 2017]. Such explanations are seen as a solution to a *model reconciliation problem* (referred to as MRP) and this approach postulates that the goal of an explanation is to update the model of the observer so they can correctly evaluate the plans in question.

Similar to MRP, we can also see our explanations as model updates, but we focus on a specific type of update, namely model concretization. Unlike MRP we do not make any assumptions about the availability of human model or the human's computational capabilities. The assumption that we have access to foils help us scale to much larger problems as compared to the original MRP approach to generate contrastive explanations. Following the conventions of the original MRP paper, we can see that the explanations studied here are both complete and monotonic.

As noted, our work is closely related to the well studied method of counter-example guided refinement or CEGAR that was originally developed for Model checking. Many planning works have successfully used CEGAR based methods to generate heuristics for plan generation ([Seipp and Helmert, 2013; 2014]). The idea of foil resolution set for a given concretization is also closely related to the process of identifying spurious counter examples employed by CEGAR based methods (c.f [Haslum *et al.*, 2012; Keyder *et al.*, 2012; Steinmetz and Hoffmann, 2016]). One major difference between our work and standard CEGAR based methods is the fact that in our setting the abstract model producing the foil (or counter-example) is unknown. Since we are exclusively dealing with spurious counter-examples we are also not bound to testing our foils (in other words identifying faults or pivot states) in the most concrete model (which could be quite expensive). Further, traditional CEGAR methods are generally not as focused on identifying the cheapest refinements.

Domain Name	$C_{\mathbb{P}}$	$\lvert\mathbb{P}\rvert$	$\lvert F\rvert$	Blind Search (Optimal)			Heuristic Search			Greedy Set Cover		
				Cost	Size	Time(S)	Cost	Size	Time(S)	Cost	Size	Time(S)
Barman	84.07	7	1	6.87	1	2.43	6.87	1	2.08	6.87	1	3.61
	84	7	2	8.94	1.22	6.35	8.94	1.22	5.71	9.90	1.39	6.05
	90.7	7	4	17.19	1.77	24.99	17.19	1.77	23.7	18.45	1.97	10.34
Rover	168.66	12	1	3.58	1	7.86	3.58	1	5.22	3.58	1	19.18
	188.83	12	2	6.13	1.48	51.36	6.12	1.48	34.04	6.26	1.52	30.5
	192.83	12	4	10.87	2	203.83	10.87	2	181.87	11.42	2.19	49.32
Satellite	53.01	4	1	18.73	1	2.23	18.73	1	1.92	18.73	1	1.49
	60.77	4	2	32	1.61	7.21	32	1.6	5.86	32.53	1.7	3.04
	62.73	4	4	43.27	2.29	18.67	43.27	2.29	16.42	43.88	2.39	5.85
Woodworking	156.71	7	1	14.45	1	2.84	14.45	1	2.23	14.45	1	3.35
	146.33	7	2	20.62	1.21	6.88	20.62	1.21	4.93	21.38	1.38	6.25
	154	7	4	28.62	1.69	24.70	28.62	1.69	19.49	30.41	2	12.13
Sokoban	220.6	3	1	51.21	1	1.51	51.21	1	1.35	51.21	1	1.28
	151.72	3	2	94.52	1.55	3.93	94.52	1.55	3.35	98.31	1.73	2.59
	220.69	3	4	136.41	2.22	8.75	136.41	2.22	8.3	141.93	2.37	5.23

Figure 4: Table showing runtime/cost for explanations generated for standard IPC domains. Column $\lvert\mathbb{P}\rvert$ represents number of predicates that were used in generating the lattice, while $C_{\mathbb{P}}$ represents the cost of an explanation that tries to concretize all propositions in $\mathbb{P}$ and provides an upper bound on explanation cost.

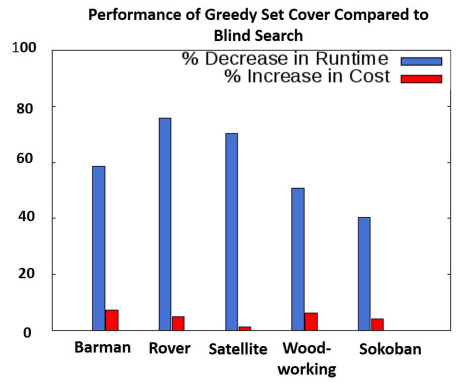

Figure 5: The graph compares the performance of greedy set cover against the optimal blind search for $\lvert F\rvert = 4$. It plots the average time saved by the set cover and the average increase in cost of the solution for each domain.

Many abstraction schemes have been proposed for planning tasks (starting with [Sacerdoti, 1974]), but in this paper, we mainly focused on state abstractions and based our formulation on previous works like [Srivastava *et al.*, 2016] and [Backstrom and Jonsson, 2013]. It would be interesting to see how we can extend the approaches discussed in this paper to handle temporal and procedural abstractions (e.g., HLAs [Marthi *et al.*, 2007]).

6 Conclusion and Discussion

In this paper, we investigated the problem of generating explanations when the explainee understands the task model at a lower levels of abstraction. We looked at how we can use explanations as concretization for such scenarios and proposed algorithms for generating minimal explanations. One unique aspect of our approach is the use of foils as a way of capturing human confusion about the problem. This not only helps us formulate more efficient explanation generation methods but also aligns with the widely held belief that human expect contrastive explanations (c.f. [Lombrozo, 2012; 2006]). Moreover, in most real-world scenarios humans usually include the foil in the request for explanations unless the foil is quite apparent from the context. Future directions include extending the methods to handle models that are incorrect in addition to being imprecise and looking at other possible methods for abstraction. We also plan to perform human factors studies on this explanation paradigm to evaluate its effectiveness.

Acknowledgments

We thank Dan Weld for helpful comments on a previous draft. This research is supported in part by the AFOSR grant FA9550-18-1-0067, ONR grants N00014161-2892, N00014-13-1-0176, N00014- 13-1-0519, N00014-15-1-2027, and the NASA grant NNX17AD06G.

References

[Backstrom and Jonsson, 2013] Christer Backstrom and Peter Jonsson. Bridging the gap between refinement and heuristics in abstraction. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[Bercher *et al.*, 2014] Pascal Bercher, Susanne Biundo, Thomas Geier, Thilo Hoernle, Florian Nothdurft, Felix Richter, and Bernd Schattenberg. Plan, repair, execute, explain-how planning helps to assemble your home theater. In *ICAPS*, 2014.

[Bernhard and Vygen, 2008] Korte Bernhard and J Vygen. Combinatorial optimization: Theory and algorithms. *Springer, Third Edition, 2005.*, 2008.

[Chakraborti *et al.*, 2017] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Plan

explanations as model reconciliation: Moving beyond explanation as soliloquy. In *IJCAI*, 2017.

[Clarke *et al.*, 2000] Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement. In *International Conference on Computer Aided Verification*, pages 154–169. Springer, 2000.

[Culberson and Schaeffer, 1998] Joseph C Culberson and Jonathan Schaeffer. Pattern databases. *Computational Intelligence*, 14(3):318–334, 1998.

[Edelkamp, 2000] Stefan Edelkamp. Planning with pattern databases. In *ECP*, 2000.

[Fox *et al.*, 2017] Maria Fox, Derek Long, and Daniele Magazzeni. Explainable Planning. In *IJCAI XAI Workshop*, 2017.

[Hart *et al.*, 1968] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[Haslum *et al.*, 2012] Patrik Haslum, John Slaney, Sylvie Thiébaux, et al. Incremental lower bounds for additive cost planning problems. In *ICAPS*, volume 12, pages 74–82, 2012.

[International Planning Competition, 2011] International Planning Competition. IPC Competition Domains. `https://goo.gl/i35bxc`, 2011.

[Kambhampati, 1990] Subbarao Kambhampati. A classification of plan modification strategies based on coverage and information requirements. In *AAAI 1990 Spring Symposium on Case Based Reasoning*. Citeseer, 1990.

[Keyder *et al.*, 2012] Emil Ragip Keyder, Jörg Hoffmann, Patrik Haslum, et al. Semi-relaxed plan heuristics. In *ICAPS*, 2012.

[Langley *et al.*, 2017] Pat Langley, Ben Meadows, Mohan Sridharan, and Dongkyu Choi. Explainable Agency for Intelligent Autonomous Systems. In *AAAI/IAAI*, 2017.

[Lombrozo, 2006] Tania Lombrozo. The structure and function of explanations. *Trends in Cognitive Sciences*, 10(10):464 – 470, 2006.

[Lombrozo, 2012] Tania Lombrozo. Explanation and abductive inference. *Oxford handbook of thinking and reasoning*, pages 260–276, 2012.

[Marthi *et al.*, 2007] Bhaskara Marthi, Stuart J Russell, and Jason Andrew Wolfe. Angelic semantics for high-level actions. In *ICAPS*, pages 232–239, 2007.

[Meadows *et al.*, 2013] Ben Leon Meadows, Pat Langley, and Miranda Jane Emery. Seeing beyond shadows: Incremental abductive reasoning for plan understanding. In *AAAI Workshop: Plan, Activity, and Intent Recognition*, volume 13, page 13, 2013.

[Miller, 2017] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *CoRR*, abs/1706.07269, 2017.

[Sacerdoti, 1974] Earl D Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial intelligence*, 5(2):115–135, 1974.

[Seegebarth *et al.*, 2012] Bastian Seegebarth, Felix Müller, Bernd Schattenberg, and Susanne Biundo. Making hybrid plans more clear to human users-a formal approach for generating sound explanations. In *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.

[Seipp and Helmert, 2013] Jendrik Seipp and Malte Helmert. Counterexample-guided cartesian abstraction refinement. In *ICAPS*, 2013.

[Seipp and Helmert, 2014] Jendrik Seipp and Malte Helmert. Diverse and additive cartesian abstraction heuristics. In *ICAPS*, 2014.

[Sohrabi *et al.*, 2011] Shirin Sohrabi, Jorge A Baier, and Sheila A McIlraith. Preferred explanations: Theory and generation via planning. In *AAAI*, 2011.

[Srivastava *et al.*, 2016] Siddharth Srivastava, Stuart J Russell, and Alessandro Pinto. Metaphysics of planning domain descriptions. In *AAAI*, pages 1074–1080, 2016.

[Steinmetz and Hoffmann, 2016] Marcel Steinmetz and Jörg Hoffmann. Towards clause-learning state space search: Learning to recognize dead-ends. In *AAAI*, pages 760–768, 2016.

[Young, 2008] Neal E Young. Greedy set-cover algorithms. In *Encyclopedia of algorithms*, pages 1–99. Springer, 2008.