A Logic-based Explanation Generation Framework for Classical and Hybrid Planning Problems (Extended Abstract) *

Stlylianos Loukas Vasileiou[1] , **William Yeoh**[1] , **Son Tran**[2] , **Ashwin Kumar**[1] , **Michael Cashmore**[3] and **Daniele Magazzeni**[4]
[1]Washington University in St. Louis
[2]New Mexico State University
[3]University of Strathclyde
[4]King's College London
{v.stylianos, wyeoh, ashwinkumar}@wustl.edu, tson@cs.nmsu.edu, michael.cashmore@strath.ac.uk, daniele.magazzeni@kcl.ac.uk

Abstract

In human-aware planning systems, a planning agent might need to explain its plan to a human user when that plan appears to be non-feasible or sub-optimal. A popular approach, called *model reconciliation*, has been proposed as a way to bring the model of the human user closer to the agent's model. In this paper, we approach the model reconciliation problem from a different perspective, that of *knowledge representation and reasoning*, and demonstrate that our approach can be applied not only to classical planning problems but also *hybrid systems planning* problems with durative actions and events/processes.

1 Introduction

From its inception, *Explainable AI Planning* (XAIP) [Fox *et al.*, 2017; Kambhampati, 2019] has garnered increasing interest due to its role in designing explainable agents that bridge the gap between theoretical and algorithmic planning literature and real-world applications [Sreedharan *et al.*, 2020]. The primary motivation of XAIP systems has been revolving around creating well integrated pipelines that can generate explanations for a given planning problem, such as explaining the optimality of a given plan. An ideal XAIP pipeline typically consists of two main components: (*i*) *Explanation generation*; and (*ii*) *Explanation communication*.

When designing XAIP systems, one of the main considerations, particularly in the explanation generation component, is taking into account the persona of the explainee [Langley, 2019]. While there can be various personas, the end user has gained significant focus as users often come with preconceived notions and expectations that may differ from the agent's outcomes. In this context, the *model reconciliation problem* (MRP) has emerged as a popular paradigm that utilizes the theory of mind [Premack and Woodruff, 1978] and considers the user's mental model during the explanation generation process of the agent [Chakraborti *et al.*, 2017].[1] Explanations in MRP aim to bring the user's model closer to the agent's model by transferring a minimum number of updates. However, most works in the MRP literature employ automated planning approaches and have been applied to classical planning problems only [Sreedharan *et al.*, 2020].

To that extent, in this paper we are mainly interested in the *explanation generation* component of XAIP, specifically through the lens of MRP, where we approach it from a different perspective – one based on *knowledge representation and reasoning*. In particular, we propose a logic-based framework for explanation generation, where given a knowledge base KB_a (of an agent) and a knowledge base KB_h (of a human user), each encoding their knowledge of a planning problem, and that KB_a entails a query q (e.g., that a proposed plan of the agent is valid or that the proposed plan is optimal), the goal is to identify an explanation $\epsilon \subseteq KB_a$ such that when it is used to update KB_h, then the updated KB_h also entails q. We then demonstrate that our approach can be applied not only to classical planning problems but also hybrid systems planning problems with durative actions, processes, and events. More specifically,

- We formally define the notion of logic-based explanations in the context of model reconciliation problems.
- We introduce a number of cost functions that can be used to reflect preferences between explanations.
- We present algorithms to compute explanations for both classical and hybrid systems planning problems.
- We empirically evaluate their performance against the current state of the art [Chakraborti *et al.*, 2017] on classical planning problems as well as provide results on hybrid systems planning problems.

In summary, our proposed framework advances the state of the art in model reconciliation approaches for explanation generation within XAIP along two key dimensions: (1) It improves the scalability for some types of classical planning problems; and (2) It generalizes the model reconciliation ap-

[*] This paper is an abridged version of our journal article [Vasileiou *et al.*, 2022].

[1] A mental model is just the user's version of the problem, which can be expressed as a graph, a planning model, or a logical knowledge base.

proach such that it can be applied to other types of planning problems beyond classical planning.

While this paper illustrates the main ideas and results of our framework, a detailed exposition (including the importance of logic in explainability and how to communicate explanations to human users) can be found in the extended version [Vasileiou *et al.*, 2022].

2 Explanation Generation Framework

We introduce the notion of an *explanation* in the following setting, where, for brevity, we use the term $\models^x_L$ for $x \in \{s, c\}$ to refer to skeptical (s) or credulous (c) entailment:

> **Explanation Generation Problem:** Given two knowledge bases KB_a and KB_h and a formula φ in a logic L, where $KB_a \models^x_L \varphi$ and $KB_h \not\models^x_L \varphi$, the goal is to identify an explanation (i.e., a set of formulae) $\epsilon \subseteq KB_a$ such that when it is used to *update* KB_h to $\widehat{KB}^{\epsilon}_h$, the updated $\widehat{KB}^{\epsilon}_h \models^x_L \varphi$.

When updating a knowledge base KB with an explanation ϵ, the updated knowledge base $KB \cup \epsilon$ may be inconsistent as there may be contradictory formulae in KB and ϵ. As such, to make the knowledge base consistent again, one needs to remove this set of contradictory formulae $\gamma \subseteq KB$ from KB. More formally:

Definition 1 (Knowledge Base Update)**.** *Given a knowledge base KB and an explanation ϵ, the updated knowledge base is $\widehat{KB}^{\epsilon} = KB \cup \epsilon \setminus \gamma$, where $\gamma \subseteq KB \setminus \epsilon$ is a set of formulae that must be removed from KB such that the updated $\widehat{KB}^{\epsilon}$ is consistent.*[2]

We now define the notion of a *support* of a formula w.r.t. a knowledge base KB before defining the notion of *explanations*.

Definition 2 (Support)**.** *Given a knowledge base KB and a formula φ in a logic L, where $KB \models^x_L \varphi$, $\epsilon \subseteq KB$ is a* support *of φ w.r.t. KB if $\epsilon \models^x_L \varphi$. Assume that ϵ is a support of φ w.r.t. KB. We say that $\epsilon \subseteq KB$ is a* $\subseteq$-minimal support *of φ if no proper sub-theory of ϵ is a support of φ. Furthermore, ϵ is a* $\lhd$-general support *of φ if there is no support ϵ' of φ w.r.t. KB such that ϵ subsumes ϵ'.*

Definition 3 (Explanation)**.** *Given two knowledge bases KB_a and KB_h and a formula φ in a logic L, where $KB_a \models^x_L \varphi$ and $KB_h \not\models^x_L \varphi$, an* explanation for φ from KB_a for KB_h *is a support ϵ w.r.t. KB_a for φ such that the updated knowledge base $\widehat{KB}^{\epsilon}_h \models^x_L \varphi$, where $\widehat{KB}^{\epsilon}_h$ is updated according to Definition 1.*

Example 1. *Consider propositional logic theories over the set of propositions $\{a, b, c\}$ with the usual definition of models, satisfaction, etc. Assume $KB_a = \{a, b, a \rightarrow c, a \wedge b \rightarrow c\}$ and $KB_{h_1} = \{a\}$. We have that $\epsilon_1 = \{a, a \rightarrow c\}$ and $\epsilon_2 = \{a, b, a \wedge b \rightarrow c\}$ are two $\subseteq$-minimal supports of c w.r.t. KB_a. Only ϵ_1 is a $\lhd$-general support of c w.r.t. KB_a since $\epsilon_2 \lhd \epsilon_1$. Both ϵ_1 and ϵ_2 can serve as explanations for c from KB_a for KB_{h_1}. Of course, KB_a is itself an explanation for c from KB_a for KB_{h_1}.*

Now consider $KB_{h_2} = \{a, \neg b\}$. In this case, both ϵ_1 and ϵ_2 are possible explanations for c from KB_a for KB_{h_2}, but if ϵ_2 is chosen, then $\neg b$ will need to be removed from KB_{h_2} so that it is consistent according to Definition 1.

2.1 Preferred Explanations

When considering explanatory systems, a natural question that potentially arises would be: *Are all explanations equal*? For example, one would want to differentiate between *trivial* and *non-trivial* explanations. While it might be acceptable in some cases, trivial explanations,[3] which are akin to a parent providing the explanation "because I said so" when asked "why?" by their child, are not preferred in most cases.

Besides computing an explanation ϵ, the agent also needs to present that explanation to the user or, in other words, describe the content of the explanation ϵ to the user. Given knowledge bases KB_a and KB_h and a formula φ, there might be several explanations for φ from KB_a for KB_h. Therefore, an agent might prefer an explanation that requires the least amount of effort[4] in presenting explanation ϵ to the human. One way to characterize the effort of the agent when presenting an explanation is to associate a cost to the elements of explanation ϵ. For example, one might prefer a subset-minimal explanation or a shortest length explanation over others. Next, we quantify the cost of an explanation, which is then used in to define a general preference relation over explanations.

We assume a cost function $\mathcal{C}_L$ that maps knowledge bases and sets of explanations to non-negative real values:

$$\mathcal{C}_L : KB_L \times \Omega \rightarrow \mathcal{R}^{\geq 0} \tag{1}$$

where Ω is the set of explanations and $\mathcal{R}^{\geq 0}$ denotes the set of non-negative real numbers. Intuitively, this function can be used to characterize different complexity measurements of an explanation. A cost function $\mathcal{C}_L$ is *monotonic* if for any two explanations $\epsilon_1 \subseteq \epsilon_2$, $\mathcal{C}_L(KB, \epsilon_1) \leq \mathcal{C}_L(KB, \epsilon_2)$. $\mathcal{C}_L$ induces a preference relation $\prec_{KB}$ over explanations as follows.

Definition 4 (Preferred Explanation)**.** *Given a cost function $\mathcal{C}_L$, a knowledge base KB_h, and two explanations ϵ_1 and ϵ_2 for KB_h, explanation ϵ_1 is* preferred *over explanation ϵ_2 w.r.t. KB_h (denoted by $\epsilon_1 \preceq_{KB_h} \epsilon_2$) iff*

$$\mathcal{C}_L(KB_h, \epsilon_1) \leq \mathcal{C}_L(KB_h, \epsilon_2) \tag{2}$$

and ϵ_1 is strictly preferred *over ϵ_2 w.r.t. KB_h (denoted by $\epsilon_1 \prec_{KB_h} \epsilon_2$) if*

$$\mathcal{C}_L(KB_h, \epsilon_1) < \mathcal{C}_L(KB_h, \epsilon_2) \tag{3}$$

[2] Intuitively, one should prefer the set of formula γ that is removed to be as small as possible, though we chose to not require such a restriction here.

[3] There might be cases where we need to explain an assumption or a fact that is missing from a KB, and therefore, trivial explanations will be succinct and acceptable.

[4] The term "effort" can refer to the effort required by the agent to present the explanation, the effort required by the human to understand the explanation, or both. For instance, the length of the explanation can serve as a measure of the effort required by both the agent and the human.

This allows us to compare explanations as follows.

Definition 5 (Most Preferred Explanation)**.** *Given a cost function $\mathcal{C}_L$ and a knowledge base KB_h, an explanation ϵ is a* most preferred explanation *w.r.t.* KB_h *if there exists no other explanation ϵ' such that $\epsilon' \prec_{KB_h} \epsilon$.*

There are several natural monotonic cost functions. For example:

- $C^1_L(KB_h, \epsilon) = |\epsilon|$, the cardinality of ϵ, indicates the number of formulae that need to be explained;
- $C^2_L(KB_h, \epsilon) = |\epsilon \setminus KB_h|$, the cardinality of $\epsilon \setminus KB_h$, indicates the number of *new* formulae that need to be explained;
- $C^3_L(KB_h, \epsilon) = \mathit{length}(\epsilon)$ indicates the number of literals in ϵ that need to be explained.

2.2 Explanations in Planning Problems

Classical and hybrid planning problems can be encoded as SAT [Kautz and Selman, 1992] and SMT problems [Cashmore *et al.*, 2020], respectively. As such, our logic-based notions of explanations proposed in the previous section can be applied to explainable planning, particularly the model reconciliation problem, in the context of explaining classical and hybrid planning problems. Nonetheless, a model reconciliation problem has been strictly defined for explaining optimal plans [Chakraborti *et al.*, 2017]. We can, however, relax this definition and generalize it for arbitrary, valid plans. The reason is that, even if optimality cannot be guaranteed, the human user may have doubts about the validity of a plan (i.e., whether the plan is sound and can be executed to achieve the goal). Therefore, valid plan explanations are crucial for engendering trust in the user.

We focus on the following two problems: (1) Explaining the *validity* of a plan to the user, and (2) Explaining the *optimality* of a plan to the user, where we define them using logical notations.

Plan Validity

Assume π is a valid plan with respect to KB_a but not KB_h. In other words, it is not possible to execute π to achieve the goal with respect to KB_h. For example, an action in the plan cannot be executed because its precondition is not satisfied, an action in the plan does not exist, or the goal is not reached after the last action in the plan is executed. From the perspective of logic, a plan is valid if there exists at least one model in KB_h in which the plan can be executed and the goal is reached:

Definition 6 (Plan Validity)**.** *Given a planning problem Π, a plan π of Π, where α_t is an action of the plan at time step t, and a knowledge base KB_h encoding Π, π is a* valid *plan in KB_h if $KB_h \models^c_L \pi \wedge g_n$, where g_n is the fact corresponding to the goal of the planning problem at time step n.*

Plan Optimality

Assume that π^* is an optimal plan in a model of KB_a. To explain the optimality of π^* to KB_h, we need to prove that no shorter (optimal) plan exists in KB_h. Thus, we need to prove that no shorter plan exists in *all* models of KB_h. This can be easily done by using the notion of skeptical entailment.

Definition 7 (Plan Optimality)**.** *Given a planning problem Π, a plan π of Π with length n, and a knowledge base KB_h encoding Π, the plan π is* optimal *in KB_h if and only if $KB_h \models^c_L \pi \wedge g_n$* and *$KB_h \models^s_L \phi$, where $\phi = \bigwedge_{t=0}^{n-1} \neg g_t$ and g_t is the fact corresponding to the goal of the planning problem at time step t.*

In essence, the query ϕ in the above definition is that no plan of lengths 1 to $n-1$ exists. Therefore, when combined with the fact that a plan π of length n that achieves the goal state exists, then that plan must be an optimal plan.

Note that the Definition 7 applies only to classical planning problems and not hybrid planning problems. The reason is because the cost of a hybrid plan depends on a user-specified plan metric, and this cost is not explicitly encoded by SMT encodings of hybrid plans. Nonetheless, we do not view this as a significant loss since finding optimal hybrid plans is often highly intractable [Helmert, 2002].

3 Experimental Evaluation

We now describe some empirical evaluations for finding explanations on classical and hybrid planning problems, encoded as SAT and SMT problems, respectively. For a description of the exact algorithms for computing explanations, please refer to the extended paper [Vasileiou *et al.*, 2022].

Setup and Prototype Implementation: The experiments were run on a Macbook Pro comprising an Intel Core i7 2.6GHz processor with 16GB of memory. The knowledge bases representing the planning problems were each encoded up to the time step that the optimal (or valid) plan was found. To encode the knowledge bases for classical planning problems, we used our own implementation of the encoding by [Kautz *et al.*, 1996], whereas for hybrid planning problems we used the encoding provided in SMTPLAN [Cashmore *et al.*, 2016]. The time limit for all experiments was set to 1500s. We have also made our source code available in a publicly-accessible repository.[5]

3.1 Efficacy on Classical Planning Problems

In this set of experiments, we examine the performance of our approach, referred to as LOGIC, for finding most-preferred explanations for plan validity and optimality on classical planning benchmarks from the International Planning Competition (IPC).[6] As a baseline, we used the current planning-based state-of-the-art algorithm by [Chakraborti *et al.*, 2017], referred to as CSZK – the initials of the last names of the authors.[7] We used the explanation length $|\epsilon|$ as the cost function of the algorithms.

We used the actual IPC instances as the model of the agent (i.e., KB_a), and tweaked that model and assigned it to be the model of the human user (i.e., KB_h). In order to make a more comprehensive analysis, we considered five different

[5] https://github.com/YODA-Lab/Explanation-Generation-for-Planning-Problems.

[6] https://github.com/potassco/pddl-instances.

[7] We used the implementation of the authors, which is publicly available at https://github.com/TathagataChakraborti/mmp.

Prob.		Scenario 1			Scenario 2			Scenario 3			Scenario 4			Scenario 5		
		$\|\epsilon\|$	CSZK	LOGIC	$\|\epsilon\|$	CSZK	LOGIC	$\|\epsilon\|$	CSZK	LOGIC	$\|\epsilon\|$	CSZK	LOGIC	$\|\epsilon\|$	CSZK	LOGIC
BLOCKS-WORLD	4	1	**0.5s**	3.0s	2	**0.5s**	0.7s	3	2.0s	**1.5s**	3	32.0s	**16.0s**	2	–	**0.7s**
	5	2	**2.5s**	8.5s	3	**2.0s**	2.5s	5	17.0s	**6.0s**	7	–	**194.0s**	4	–	**2.0s**
	6	1	**1.0s**	25.0s	2	**0.5s**	5.5s	3	**3.0s**	6.0s	4	213.0s	**120.0s**	5	–	**5.0s**
	8	3	**62.0s**	297.0s	3	**1.0s**	29.5s	6	869.0s	**30.0s**	7	–	**203.0s**	5	–	**27.0s**
ELEV-ATOR	1	1	0.5s	**0.1s**	2	1.0s	**0.1s**	2	0.5s	**0.1s**	2	1.0s	**0.1s**	1	–	**0.1s**
	10	2	1.5s	**0.7s**	2	**0.5s**	**0.5s**	3	3.0s	**0.4s**	4	57.0s	**2.5s**	6	–	**0.2s**
	15	2	**1.5s**	3.0s	2	**1.0s**	13.0s	3	3.0s	**2.0s**	4	57.0s	**10.0s**	6	–	**1.2s**
	19	2	**2.0s**	8.0s	2	**0.5s**	25.0s	3	**2.5s**	10.0s	4	49.0s	**20.0s**	14	–	**5.0s**
ROVER	1	1	**0.5s**	6.0s	2	**0.5s**	7.0s	4	33.0s	**5.0s**	6	–	**5.0s**	4	–	**1.5s**
	2	1	**1.0s**	4.0s	1	**0.5s**	4.0s	4	39.0s	**4.5s**	6	–	**4.5s**	4	–	**1.3s**
	3	1	**0.5s**	7.0s	2	**0.5s**	7.5s	4	35.0s	**7.0s**	6	–	**10.0s**	6	–	**1.5s**
	4	1	**0.5s**	4.0s	1	**0.5s**	4.0s	2	**1.5s**	4.5s	4	–	**4.5s**	10	–	**5.5s**
GRIPPER	1	1	**0.5s**	1.5s	2	**0.3s**	3.0s	3	**1.5s**	40.0s	5	70.0s	**45.0s**	4	–	**2.0s**
	2	1	**0.5s**	5.0s	2	**0.8s**	7.0s	3	**2.0s**	45.0s	5	73.0s	**49.0s**	5	–	**6.0s**
	3	1	**0.7s**	5.0s	2	**1.0s**	7.0s	3	**2.5s**	45.0s	5	163.0s	**60.0s**	8	–	**15.0s**
	4	1	**1.5s**	38.0s	2	**3.0s**	50.0s	3	**5.0s**	50.0s	5	–	**80.0s**	11	–	**28.0s**

Table 1: Evaluation of our approach LOGIC and CSZK on Varying PDDL Domains and Scenarios.

ways to tweak the models, resulting in the following five scenarios.

- **Scenario 1:** We removed one random precondition from every action in the human's model.
- **Scenario 2:** We removed one random effect from every action in the human's model.
- **Scenario 3:** We removed one random precondition and one random effect from every action in the human's model.
- **Scenario 4:** We removed (on average) fifteen random preconditions and effects from every action in the human's model.
- **Scenario 5:** We removed (on average) ten random predicates from the initial states in the human's model.

Table 1 tabulates the length of the explanations $|\epsilon|$ as well as the runtimes of LOGIC and CSZK. We did not report runtimes of CSZK for Scenario 5 as the available implementation could not handle that scenario. In general, CSZK outperformed LOGIC in a majority of cases, except for Scenarios 3 and 4 in all domains. These cases also happen to be the cases where the explanation length $|\epsilon|$ is larger. The reason is that CSZK needs to search over a larger search space as the explanation length increases. As such, its runtime also increases. In contrast, the runtimes of LOGIC remain relatively unchanged with varying explanation lengths. The reason is that the runtimes of LOGIC are dominated by the size of the encoded knowledge bases, which are independent of the explanation lengths.

It is important to note that Vasileiou *et al.* [2021] proposed a more efficient approach for computing minimal logic-based explanations in model reconciliation problems.

3.2 Efficacy on Hybrid Planning Problems

In this set of experiments we investigate the generality of our approach on hybrid planning problems, and specifically, on plan validity. We consider the same scenarios as in the previous section.

Table 2 tabulates the results. Overall, LOGIC was able to maintain small runtimes of less than $1s$ in the majority of instances. This is due to the fact that the size of the encoded knowledge bases are relatively small because SMT-PLAN uses the iterative encoding facility of the $z3$ solver [De Moura and Bjørner, 2008]. Particularly, the encoding of each layer consists of the following steps: Adding the new variables and constraints for the next happening, adding the goal constraints to the new constraint set, pushing the constraint set onto the stack, solving, and popping the goal constraint set off the stack. As such, at each step in the iterative deepening with $z3$, only the latest layer needs to be encoded. These results demonstrate that our approach can be generalized beyond classical planning to hybrid planning, improving the applicability of explainable planning approaches.

Prob.		Scenario 1		Scenario 2		Scenario 3		Scenario 4		Scenario 5	
		$\|\epsilon\|$	LOGIC	$\|\epsilon\|$	LOGIC	$\|\epsilon\|$	LOGIC	$\|\epsilon\|$	LOGIC	$\|\epsilon\|$	LOGIC
LINEAR GENER.	1	0	0.1s	2	0.1s	2	0.2s	1	0.1s	2	0.1s
	3	0	0.1s	2	0.2s	2	0.8s	1	0.2s	2	0.2s
	5	0	0.2s	2	0.2s	2	2.0s	1	0.4s	2	0.4s
	7	0	0.3s	2	0.5s	2	4.0s	1	1.0s	2	0.6s
TORI-CELLI	1	1	0.2s	2	0.3s	2	0.4s	3	0.6s	4	0.2s
	2	1	0.4s	2	1.3s	2	2.0s	3	1.1s	5	0.9s
	3	1	0.5s	2	5.0s	2	11.0s	3	2.8s	7	3.6s
	4	1	1.0s	2	16.0s	2	38.0s	3	5.8s	5	1.1s
GENER. EVENTS	1	1	0.2s	2	0.2s	3	2.5s	3	0.2s	2	0.2s
	2	1	0.3s	2	0.5s	3	5.0s	3	0.5s	3	0.2s
	3	2	0.8s	2	1.3s	3	10.0s	3	1.5s	4	0.7s
	4	1	1.3s	2	2.0s	3	26.0s	3	2.5s	6	0.9s
CAR NO DRAG	1	2	0.2s	1	0.3s	3	0.3s	3	0.3s	2	0.2s
	2	2	0.2s	1	0.2s	2	0.4s	3	0.3s	3	0.3s
	3	2	0.3s	1	0.3s	2	0.2s	3	0.4s	1	0.1s
	4	2	0.2s	1	0.2s	3	0.3s	3	0.3s	2	0.2s

Table 2: Evaluation of our approach LOGIC on Varying PDDL+ Domains and Scenarios.

4 Discussion and Conclusions

When designing explanatory systems, a question that often arise is how to identify, represent, and provide explanations. There is a general belief that logic-based systems are well equipped to address this question. For example, logic-based models such as decision trees produce explanations stemming directly from the model [Lakkaraju *et al.*, 2016; Ignatiev *et al.*, 2018]. In this work, we examined and evaluated this belief by creating a logic-based explanation generation framework for classical and hybrid planning problems for the model reconciliation problem. In this context, we made the following contributions: (1) We approached the MRP problem from the perspective of *knowledge representation and reasoning* by proposing the notion of explanations and defined plan validity and optimality in terms of knowledge bases; (2) We proposed several complexity cost functions to reflect preferences between explanations; (3) We developed algorithms for computing most-preferred explanations for plan validity and optimality; and (4) We empirically showed that our approach complements the current state of the art and is able to generalize beyond classical planning to hybrid planning.

Acknowledgments

Stylianos Loukas Vasileiou, William Yeoh, and Ashwin Kumar are partially supported by the National Science Foundation (NSF) under award 1812619. Son Tran is partially supported by NSF under awards 1757207, 1812628, and 1914635. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the United States government.

References

[Cashmore *et al.*, 2016] Michael Cashmore, Maria Fox, Derek Long, and Daniele Magazzeni. A compilation of the full PDDL+ language into SMT. In *ICAPS*, pages 79–87, 2016.

[Cashmore *et al.*, 2020] Michael Cashmore, Daniele Magazzeni, and Parisa Zehtabi. Planning for hybrid systems via satisfiability modulo theories. *Journal of Artificial Intelligence Research*, 67:235–283, 2020.

[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*, pages 156–163, 2017.

[De Moura and Bjørner, 2008] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *TACAS*, pages 337–340, 2008.

[Fox *et al.*, 2017] Maria Fox, Derek Long, and Daniele Magazzeni. Explainable planning. *arXiv preprint arXiv:1709.10256*, 2017.

[Helmert, 2002] Malte Helmert. Decidability and undecidability results for planning with numerical state variables. In *AIPS*, pages 44–53, 2002.

[Ignatiev *et al.*, 2018] Alexey Ignatiev, Filipe Pereira, Nina Narodytska, and Joao Marques-Silva. A SAT-based approach to learn explainable decision sets. In *IJCAR*, pages 627–645, 2018.

[Kambhampati, 2019] Subbarao Kambhampati. Synthesizing explainable behavior for human-AI collaboration. In *AAMAS*, pages 1–2, 2019.

[Kautz and Selman, 1992] Henry Kautz and Bart Selman. Planning as satisfiability. In *ECAI*, pages 359–363, 1992.

[Kautz *et al.*, 1996] Henry Kautz, David McAllester, and Bart Selman. Encoding plans in propositional logic. In *KR*, pages 374–384, 1996.

[Lakkaraju *et al.*, 2016] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *ACM SIGKDD*, pages 1675–1684, 2016.

[Langley, 2019] Pat Langley. Varieties of explainable agency. In *ICAPS Workshop on Explainable AI Planning*, 2019.

[Premack and Woodruff, 1978] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978.

[Sreedharan *et al.*, 2020] Sarath Sreedharan, Tathagata Chakraborti, and Subbarao Kambhampati. The emerging landscape of explainable automated planning & decision making. In *IJCAI*, pages 4803–4811, 2020.

[Vasileiou *et al.*, 2021] Stylianos Loukas Vasileiou, Alessandro Previti, and William Yeoh. On exploiting hitting sets for model reconciliation. In *AAAI*, pages 6514–6521, 2021.

[Vasileiou *et al.*, 2022] Stylianos Loukas Vasileiou, William Yeoh, Tran Cao Son, Ashwin Kumar, Michael Cashmore, and Daniele Magazzeni. A logic-based explanation generation framework for classical and hybrid planning problems. *Journal of Artificial Intelligence Research*, 73:1473–1534, 2022.