On Generating Explainable Plans with Assumption-based Argumentation

No Author Given

No Institute Given

Abstract. Planning is a classic problem in Artificial Intelligence (AI). Recently, the need for creating "Explainable AI" has been recognised and voiced by many researchers. Leveraging on the strength of argumentation, in particular, the Related Admissible semantics for generating explanations, this work makes an initial step towards "explainable planning". We illustrate (1) how plan generation can be equated to constructing acceptable arguments and (2) how explanations for both "planning solutions" as well as "invalid plans" can be obtained by extracting information from an arguing process. We present an argumentation-based model which takes plans written in a STRIPS-like language as its inputs and returns Assumption-based Argumentation (ABA) frameworks as its outputs. The presented plan construction mapping is both sound and complete in that the planning problem has a solution if and only if its corresponding ABA framework has a set of Related Admissible arguments with the planning goal as its topic. We use the classic Tower of Hanoi puzzle as our case study and demonstrate how ABA can be used to solve this planning puzzle while giving explanations.

1 Introduction

Planning, known as the "reasoning side of acting" [16], has been long studied in artificial intelligence and seen its applications in many areas form robot navigation to manufacturing scheduling. Much research has been devoted to the development of expressive planning languages and efficient planners, e.g. [17, 19]. Recently, we see that the need for developing transparent and explainable autonomous intelligent systems has been recognised and voiced by many researchers [5, 20]. At the same time, argumentation [14], a knowledge representation and modelling technique in rapid development, for reasoning with incomplete and inconsistent information with its ability in explaining the results and processes of computation, has seen its use in many applications.

In this work, we present a study on modelling and solving planning problems with Assumption-based Argumentation (ABA) [4]. We establish the correspondence between ABA arguments and plans such that a planning problem has a solution (plan) if and only if the argument representing this solution is acceptable. On the front of plan explanation, we observe that a plan solution (1) meets all of its goals while (2) satisfying all pre-conditions of its actions. Thus, explanations for "successful" plans are focused on justifying these two criteria; and explanations for "failed" plans are focused on identifying unmet pre-conditions. The tasks of generating plans and explanations are unified under the computation of Related Admissible set of arguments using *dispute trees* [6].

To make our argumentation-based planning study concrete, we take a version of a classic planning puzzle, *Tower of Hanoi*, as a case study example. This example, though conceptually simple, exhibits typical planning characteristics and challenges. Roughly speaking, a *plan* takes a world containing multiple objects in the world's *initial state* to its *goal state* via a sequence of *actions* in discrete time steps. In each time step, multiple actions, subject to various *pre-conditions*, are possible to be performed. A search for suitable actions is needed to find solutions. By developing argumentation-based approaches to Tower of Hanoi, we establish the feasibility of using argumentation to plan and reveal the strength and potential future development for argumentation-based planning.

ABA is selected as the modelling and computation vehicle as it is a versatile structured argumentation framework with many successful applications. Roughly, ABA assumptions represent actions (in the sense that we assume actions are valid unless its pre-conditions are not met), invalidity of world states (in the sense that we assume the environment is not in a specific state unless we prove it is in). Acceptable arguments correspond to planning solutions. We use well-defined argumentation semantics with sound computation tools to generate plans and explanations.

2 Background

Assumption-based Argumentation (ABA) frameworks are tuples $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$, where

- $\langle \mathcal{L}, \mathcal{R} \rangle$ is a deductive system, with $\mathcal{L}$ the *language* and $\mathcal{R}$ a set of *rules* of the form $s_0 \leftarrow s_1, \ldots, s_m (m \geq 0, s_i \in \mathcal{L})$;
- $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set of *assumptions*;
- $\mathcal{C}$ is a total mapping from $\mathcal{A}$ into $2^{\mathcal{L}} - \{\{\}\}$, where each $s \in \mathcal{C}(a)$ is a *contrary* of a, for $a \in \mathcal{A}$.

Given $\rho = s_0 \leftarrow s_1, \ldots, s_m$, s_0 is referred to as the *head* and $s_1, \ldots, s_m$ as the *body*.

Arguments are deductions of claims using rules and supported by sets of assumptions; *Attacks* are *targeted* at the assumptions in the support of arguments:

- *an argument for (claim)* $s \in \mathcal{L}$ *supported by* $\Delta \subseteq \mathcal{A}$ (denoted $\Delta \vdash s$) is a finite tree with nodes labelled by sentences in $\mathcal{L}$ or by τ[1], the root labelled by s, leaves either τ or assumptions in Δ, and non-leaves s' with, as children, the elements of the body of some rule ρ with head s';
- *an argument* $A = \Delta_1 \vdash s_1$ attacks an argument $\Delta_2 \vdash s_2$ if and only if s_1 is a contrary of some assumption α in Δ_2, and we say A *targets at* α.

A *set of arguments* As *is admissible* if and only if As is *conflict-free* (i.e. no argument in As attacks any argument in As) and all arguments attacking some argument in As are counter attacked by arguments in As; an *argument is admissible* if and only if it belongs to an admissible set of arguments.

We will use the notion of *Related Admissible* and *Argument Explanation* introduced in [7] for some of our results, defined as follows. Given an ABA framework $F =$

[1] $\tau \notin \mathcal{L}$ represents "true" and stands for the empty body of rules.

$\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$, $\mathbb{AG}^F$ denotes the set of all arguments in F. Let $X, Y \in \mathbb{AG}^F$. X *defends* Y if and only if: (1) $X = Y$; or (2) $\exists Z \in \mathbb{AG}^F$, such that X attacks Z and Z attacks Y; or (3) $\exists Z \in \mathbb{AG}^F$, such that X defends Z and Z defends Y. $S \subseteq \mathbb{AG}^F$ *defends* $X \in \mathbb{AG}^F$ if and only if $\forall Y \in S$: Y defends X. Let $s \in \mathcal{L}$ and $A, B \in \mathbb{AG}^F$, A *defends* s if and only if s is the claim of B and A defends B.

A *set of arguments* As *is Related Admissible* if and only if: (1) As is admissible, (2) there exists a *topic* sentence χ (of As), χ is the claim of some argument in As, such that for all $B \in As$, B defends χ. As is an *explanation* of χ.

We will use the *abstract dispute trees* of [6] to compute explanations for our plans. An *abstract dispute tree* for an argument A is a (possibly infinite) tree $\mathcal{T}^a$ such that:[2]

1. every node of $\mathcal{T}^a$ holds an argument B and is labelled by either *proponent* (**P**) or *opponent* (**O**), but not both, denoted by $L : B$, for $L \in \{\text{P,O}\}$; (a node labelled by **P/O** is called a **P/O** node, respectively);
2. the root of $\mathcal{T}^a$ is a **P** node holding A;
3. for every **P** node N holding an argument B, and for every argument C that attacks B, there exists a child of N, which is an **O** node holding C;
4. for every **O** node N holding an argument B, there exists *at most*[3] one child of N which is a **P** node holding an argument which targets some assumption α in the support of B; if N has a child attacking α, then α is said to be the *culprit* in B;
5. there are no other nodes in $\mathcal{T}^a$ except those given by 1-4 above.

The set of all assumptions in (the support of arguments held by) the **P** nodes in $\mathcal{T}^a$ is called the *defence set* of $\mathcal{T}^a$. In an abstract dispute tree $\mathcal{T}^a$, a **P** node N is *defeated* if and only if N is the root of a sub-tree in $\mathcal{T}^a$ such that the defence set of the sub-tree is not admissible. A *winning attacker* N' of N is a child node of N such that either (1) there is an **O** leaf node in the tree rooted at N' or (2) there is an argument held at both a **P** node in the tree rooted at N' and an **O** node in $\mathcal{T}^a$. Abstract dispute trees can be used to compute (related) admissibility semantics:

- Let an abstract dispute tree $\mathcal{T}^a$ be *admissible* if and only if each **O** node has *exactly* one child and no culprit in the argument of an **O** node in $\mathcal{T}^a$ belongs to the defence set of $\mathcal{T}^a$. If a dispute tree is not admissible, it is *non-admissible*.
- The defence set of an admissible abstract dispute tree is admissible (Theorem 5.1 in [6]), and thus the root node of an admissible dispute tree is admissible.
- The defence set of an admissible abstract dispute tree is Related Admissible (Theorem 1 in [7]) with the claim of the argument held by the root of the tree being the topic sentence (Theorem 5 in [7]).

[2] Here, a stands for 'abstract'. Also, 'proponent' and 'opponent' should be seen as roles/fictitious participants in a debate rather than actual agents.

[3] In the original definition of abstract dispute tree [6], every **O** node is required to have *exactly* one child. We incorporate this requirement into the definition of *admissible* dispute tree given later, so that our notion of admissible abstract dispute tree and the admissible abstract dispute trees of [6] coincide.

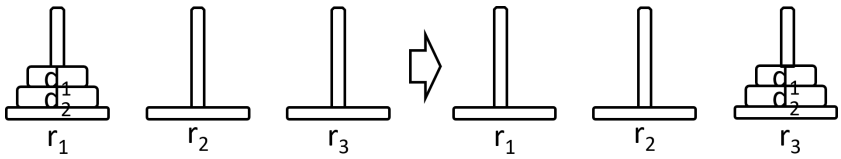

Fig. 1. A Tower of Hanoi game with two discs d_1 and d_2 and three rods r_1, r_2 and r_3. The initial state is shown on the left hand side and the goal state is shown on the right hand side.

3 Planning Preliminaries

We consider an instance of the standard, also the most widely used, planning representation, STRIPS, as given in [16]. A planning problem $\mathcal{P}$ is a tuple $\mathcal{P} = (\Sigma, s_0, S_g)$; $\Sigma = (S, A, \gamma)$ is the planning domain, S the set of states, A the set of actions, γ the deterministic transition function, s_0 the initial state, and S_g the set of goal states. Each state in S is described by a set of predicates and each predicate is either a *flexible relation* or a *rigid relation*. The transition function γ is specified through a set of *planning operators*, each representing an *action*. A planning operator is given by *name*, *precond* and *effects*, where name is syntactically a predicate, precond and effects are sets of predicates, describing the pre-conditions and the effects of the action, respectively. A plan is a sequence of actions; and a solution to a planning problem is a plan from the initial state to the goal state.

We use the following classic *Tower of Hanoi* example to illustrate.

Example 1. As in a classic Tower of Hanoi game, we have three rods, r_1, r_2, r_3. To simplify the example, we use only two disks d_1 and d_2. The problem states S is described by two flexible relations `clear` and `on`, as well as a rigid relation `smaller`. The initial and the goal states are shown in Fig. 1. Specifically, the initial state s_0 is described with the following predicates:

$\texttt{clear}(\texttt{r}_2)$ $\texttt{clear}(\texttt{r}_3)$ $\texttt{clear}(\texttt{d}_1)$ $\texttt{on}(\texttt{d}_1, \texttt{d}_2)$
$\texttt{on}(\texttt{d}_2, \texttt{r}_1)$ $\texttt{smaller}(\texttt{d}_1, \texttt{r}_1)$ $\texttt{smaller}(\texttt{d}_1, \texttt{r}_2)$ $\texttt{smaller}(\texttt{d}_1, \texttt{r}_3)$
$\texttt{smaller}(\texttt{d}_2, \texttt{r}_1)$ $\texttt{smaller}(\texttt{d}_2, \texttt{r}_2)$ $\texttt{smaller}(\texttt{d}_2, \texttt{r}_3)$ $\texttt{smaller}(\texttt{d}_1, \texttt{d}_2)$

The goal state s_g is described by: $\texttt{on}(\texttt{d}_1, \texttt{d}_2)$ $\texttt{on}(\texttt{d}_2, \texttt{r}_3)$.
There is a single planning operator:

$\texttt{move}(\texttt{D}, \texttt{A}, \texttt{B})$
  $\texttt{precond}$: $\texttt{smaller}(\texttt{D}, \texttt{B}), \texttt{on}(\texttt{D}, \texttt{A}), \texttt{clear}(\texttt{D}), \texttt{clear}(\texttt{B})$
  $\texttt{effects}$: $\texttt{clear}(\texttt{A}), \texttt{on}(\texttt{D}, \texttt{B}), \neg\texttt{on}(\texttt{D}, \texttt{A}), \neg\texttt{clear}(\texttt{B})$

The action sequence $(\texttt{move}(\texttt{d}_1, \texttt{r}_2), \texttt{move}(\texttt{d}_2, \texttt{r}_3), \texttt{move}(\texttt{d}_1, \texttt{d}_2))$ is a solution.

To model planning with argumentation, we take the *bounded planning approach* as in SAT based planners [16]. Namely, we focus on finding plans of some known length n for some fixed n. Each $i, 0 \leq i \leq n$ is a *step* of the planning problem, and *for each step k there is one and only one action taking place*.[4] Specifically,

- we denote each predicate with k variables representing a flexible relation as a new predicate with $k + 1$ variables, where the last variable is the step;

[4] This is a standard approach in planning as it allows the complete specification of the planning search space. Techniques have been developed to estimate the step bound, see e.g. [16].

– we leave all predicates representing rigid relations unchanged;
– for each action taking place at step k, its pre-conditions are composed of predicates representing rigid relations and predicates representing flexible relations with their last variables k; effects of this action are composed of predicates representing flexible relations with their last variables $k+1$.

Effectively, a bounded planning problem can be described with a tuple $\langle \mathbb{A}, s_0, s_g, \mathbb{F}, \mathbb{R}, \mathbb{K} \rangle$, a set of actions $\mathbb{A}$, the initial state s_0, the goal state s_g, a set of flexible relation $\mathbb{F}$, a set of rigid relation $\mathbb{R}$ and the step bound $\mathbb{K}$. The goal state is a set of *goals* denoted by flexible relations with their last variable (the step variable) specified as $\mathbb{K}$.

Given a plan $(m_1, \ldots, m_n)$ in a bounded planning problem, we say that a flexible relation $\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K})$[5] *holds* at step k if and only if either (1) $\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{0}) \in s_0$, or (2) $\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{i})$ in the effects of some action $m_i, i \leq k$ and $\neg\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{j})$ is not in the effects of any action m_j for $i < j \leq k$.

For each action m in a plan P, we use $S(m)$ to denote the step of action m in P.

Example 2. (Example 1 continued.) With step introduced, the three predicates are:
$\mathtt{clear}(\mathtt{X}, \mathtt{K})$, $\mathtt{on}(\mathtt{X}, \mathtt{Y}, \mathtt{K})$, $\mathtt{smaller}(\mathtt{X}, \mathtt{Y})$.
The planning operator is:

$\mathtt{move}(\mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K})$
  $\mathtt{precond}$: $\mathtt{smaller}(\mathtt{D}, \mathtt{B}), \mathtt{on}(\mathtt{D}, \mathtt{A}, \mathtt{K}), \mathtt{clear}(\mathtt{D}, \mathtt{K}), \mathtt{clear}(\mathtt{B}, \mathtt{K})$
  $\mathtt{effects}$: $\mathtt{clear}(\mathtt{A}, \mathtt{K}+1), \mathtt{on}(\mathtt{D}, \mathtt{B}, \mathtt{K}+1), \neg\mathtt{on}(\mathtt{D}, \mathtt{A}, \mathtt{K}+1), \neg\mathtt{clear}(\mathtt{B}, \mathtt{K}+1)$

Suppose that $\mathbb{K} = 3$, a plan taking the initial state

$\mathtt{clear}(\mathtt{r}_2, \mathtt{0})\ \mathtt{clear}(\mathtt{r}_3, \mathtt{0})\ \mathtt{clear}(\mathtt{d}_1, \mathtt{0})\ \mathtt{on}(\mathtt{d}_1, \mathtt{d}_2, \mathtt{0})\ \mathtt{on}(\mathtt{d}_2, \mathtt{d}_1, \mathtt{0})$

to the goal state

$\mathtt{on}(\mathtt{d}_1, \mathtt{d}_2, \mathtt{3})\ \mathtt{on}(\mathtt{d}_2, \mathtt{d}_1, \mathtt{3})$

is the sequence $(\mathtt{move}(\mathtt{d}_1, \mathtt{d}_2, \mathtt{r}_2, \mathtt{0}), \mathtt{move}(\mathtt{d}_2, \mathtt{r}_1, \mathtt{r}_3, \mathtt{1}), \mathtt{move}(\mathtt{d}_1, \mathtt{r}_2, \mathtt{d}_2, \mathtt{2}))$. Clearly, this plan is a solution to this planning problem.

4 Plan Explanations

Given a bounded planning problem introduced in the previous section, we make the following observation:

A plan is a solution if the following two conditions hold:

C1. All pre-conditions hold for all actions in the plan.
C2. All goals in the goal state hold at the end of plan.

Thus, to "explain" why a plan is a solution, we want to show that both (**C1**) and (**C2**) are satisfied in the sense that an "explanation" should justify that all pre-conditions and goals are met at the right steps, formally:

Definition 1. *Given a bounded planing problem $\mathcal{P}$ with a solution P. An explanation for P being a solution to $\mathcal{P}$ is the set $S = s_g \cup \{C | C$ is a precondition for an action in $P\}$ such that every predicate in S holds at its respective step.*

[5] We use the convention that the over-line on $\mathtt{A}$ denotes that $\overline{\mathtt{A}}$ represents a list of variables of unspecified length. Variables without over-lines are "normal" variables.

We illustrate Definition 1 with the following example.

Example 3. (Example 2 continued.) To simplify the illustration, we let the step bound $\mathbb{K} = 2$ and the goal state $s_g = \{\texttt{on}(\texttt{d}_1, \texttt{r}_2, 2), \texttt{on}(\texttt{d}_2, \texttt{r}_3, 2)\}$. To see that $(\texttt{move}(\texttt{d}_1, \texttt{r}_2, 0), \texttt{move}(\texttt{d}_2, \texttt{r}_3, 1))$ is a solution, we observe that:[6]

- all predicates in s_g hold at step 2; and
- all pre-conditions hold for $\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0)$ at step 0: $\texttt{on}(\texttt{d}_1, \texttt{d}_2, 0), \texttt{sm}(\texttt{d}_1, \texttt{r}_2), \texttt{cl}(\texttt{d}_1, 0), \texttt{cl}(\texttt{r}_2, 0)$
- all pre-conditions hold for $\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1)$ at step 1: $\texttt{on}(\texttt{d}_2, \texttt{d}_1, 1), \texttt{sm}(\texttt{d}_2, \texttt{r}_3), \texttt{cl}(\texttt{d}_2, 1), \texttt{cl}(\texttt{r}_3, 1)$

Definition 1 specifies explanations for solutions to planning problems. On the other hand, to explain "why a plan fails to be a solution", we introduce "invalidity" of plans by identifying actions not meeting their pre-conditions or occurred at the same step as other actions. These invalid actions "explain" the invalidity of a plan.

Definition 2. *Let $P = (m_1, \ldots, m_n)$ be a plan. An action m_i is* invalid *in P if either one of the following two conditions holds.*

1. *There exists a pre-condition C of m_i such that C does not hold at step $S(m_i)$. In this case, C is an* explanation *for the invalidity of m_i.*
2. *There exists an action m_j in P, $j < i$, such that $S(m_i) = S(m_j)$. In this case m_j is an* explanation *for the invalidity of m_i.*

If a plan P contains no invalid action, then P is valid*; otherwise, P is* invalid*.*

We illustrate Definition 2 with the following example.

Example 4. (Example 2 continued.) With the Tower of Hanoi game as specified, let the step bound $\mathbb{K} = 2$. Given a plan $P = (\texttt{move}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0), \texttt{move}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1))$, the action $\texttt{move}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1)$ is invalid as $\texttt{small}(\texttt{d}_2, \texttt{d}_1)$, a pre-condition of $\texttt{move}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1)$, does not hold at step 1. (Note that since $\texttt{small}(\texttt{A}, \texttt{B})$ is a rigid relation, $\texttt{small}(\texttt{d}_2, \texttt{d}_1)$ never holds.) We thus have $\texttt{small}(\texttt{d}_2, \texttt{d}_1)$ as an explanation for $\texttt{move}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1)$. Since P contains an invalid action, P is invalid.

For $P' = (\texttt{move}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0), \texttt{move}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0))$, the action $\texttt{move}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0)$ is invalid, and its explanation is $\texttt{move}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0)$. This is easy to see as both actions occur at step 0. As previously, P' is invalid.

5 Planning with ABA

Thus far, we have reviewed bounded planning problems and presented several definitions of explanations. We show how ABA can be used to solve bounded planning problems in this section and how explanations can be extracted from an arguing process in the next section. To model planning with ABA, the main task is to represent bounded planning problems with ABA frameworks. Formally,

[6] $\texttt{mv}$, $\texttt{cl}$ and $\texttt{sm}$ are short-hands for $\texttt{move}$, $\texttt{clear}$ and $\texttt{smaller}$, respectively.

Definition 3. *Given a bounded planning problem $\mathcal{P} = \langle \mathbb{A}, s_0, s_g, \mathbb{F}, \mathbb{R}, \mathbb{K} \rangle$, let the following denote a generic action in $\mathbb{A}$,*

$\mathtt{act}(\overline{\mathtt{A}}, \mathtt{K})$
  $\mathtt{precond} : \mathtt{PreC_1}(\overline{\mathtt{C_1}}, \mathtt{K}), \ldots, \mathtt{PreC_w}(\overline{\mathtt{C_w}}, \mathtt{K})$
  $\mathtt{effects} : \mathtt{Ef_1}(\overline{\mathtt{E_1}}, \mathtt{K}+1), \ldots, \mathtt{Ef_m}(\overline{\mathtt{E_m}}, \mathtt{K}+1)$,

then the ABA framework corresponding to $\mathcal{P}$ is $F = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$, in which[7]

• *$\mathcal{R}$ is constructed as follows.*

1. *Let $s_g = \{\mathtt{g_1}, \ldots, \mathtt{g_n}\}$, insert the rule:* $\mathtt{goal} \leftarrow \mathtt{g_1}, \ldots, \mathtt{g_n}$.
2. *For each action in $\mathbb{A}$, insert rules:*
   $\mathtt{aE}(\mathtt{act}, \overline{\mathtt{A}}, \mathtt{K}+1) \leftarrow \mathtt{act}(\overline{\mathtt{A}}, \mathtt{K})$*;* $\mathtt{hasAct}(\mathtt{K}) \leftarrow \mathtt{act}(\overline{\mathtt{A}}, \mathtt{K})$.
   $\mathtt{Ef_1}(\overline{\mathtt{E_1}}, \mathtt{K}) \leftarrow \mathtt{aE}(\mathtt{act}, \overline{\mathtt{A}}, \mathtt{K})$*; …;* $\mathtt{Ef_m}(\overline{\mathtt{E_m}}, \mathtt{K}) \leftarrow \mathtt{aE}(\mathtt{act}, \overline{\mathtt{A}}, \mathtt{K})$.
3. *For each flexible relation* $\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K})$ *in $\mathbb{F}$, insert a rule:*
   $\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K}+1) \leftarrow \mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K}), \mathtt{hasAct}(\mathtt{K}), \mathtt{df}(\mathtt{FR}, \overline{\mathtt{A}}, \mathtt{K}+1)$.
4. *For each rigid relation* $\mathtt{RR}(\overline{\mathtt{A}})$ *in $\mathbb{R}$, insert the rule:* $\mathtt{RR}(\overline{\mathtt{A}}) \leftarrow$.
5. *Let* $\mathtt{s_0} = \{\mathtt{P_1}(\overline{\mathtt{I_0}}, 0), \ldots, \mathtt{P_n}(\overline{\mathtt{I_n}}, 0)\}$*, insert rules* $\mathtt{P_1}(\overline{\mathtt{I_0}}, 0) \leftarrow$*; …;* $\mathtt{P_n}(\overline{\mathtt{I_n}}, 0) \leftarrow$.

• *$\mathcal{A}$ is constructed as follows.*

1. *For each action in $\mathbb{A}$, insert the following assumptions:*
   $\mathtt{act}(\overline{\mathtt{A}}, \mathtt{K}), \mathtt{not\_PreC_1}(\overline{\mathtt{C_1}}, \mathtt{K}), \ldots, \mathtt{not\_PreC_w}(\overline{\mathtt{C_w}}, \mathtt{K})$
2. *For each flexible relation* $\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K})$ *in $\mathbb{F}$, insert* $\mathtt{df}(\mathtt{FR}, \overline{\mathtt{A}}, \mathtt{K})$ *to $\mathcal{A}$.*

• *$\mathcal{C}$ is such that:*

1. *For each action let* $\mathtt{act}'$ *be the name of an action in $\mathbb{A}$, then*
   – $\mathcal{C}(\mathtt{act}(\overline{\mathtt{A}}, \mathtt{K})) = \{\mathtt{act}'(\overline{\mathtt{B}}, \mathtt{K}) \mid \mathtt{act}' \neq \mathtt{act}$ *or* $\overline{\mathtt{B}} \neq \overline{\mathtt{A}}\} \cup$
     $\{\mathtt{not\_PreC_1}(\overline{\mathtt{C_1}}, \mathtt{K}), \ldots, \mathtt{not\_PreC_w}(\overline{\mathtt{C_w}}, \mathtt{K})\}$*;*
   – *for* $\mathtt{i} = 1, \ldots, \mathtt{w}$, $\mathcal{C}(\mathtt{not\_PreC_i}(\overline{\mathtt{C_i}}, \mathtt{K})) = \{\mathtt{PreC_i}(\overline{\mathtt{C_i}}, \mathtt{K})\}$.
2. $\mathcal{C}(\mathtt{df}(\mathtt{FR}, \overline{\mathtt{A}}, \mathtt{K})) = \{\neg \mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K})\}$.

In Definition 3, reaching the goal state s_g is modelled with the rule

$$\mathtt{goal} \leftarrow \mathtt{g_1}, \ldots, \mathtt{g_n}$$

such that to reach the goal state, we need to prove each of its goals in the goal set. Then, for each action $\mathtt{act}(\overline{\mathtt{A}}, \mathtt{K})$, taking a list of variables $\overline{\mathtt{A}}$ at step $\mathtt{K}$, we use the rule

$$\mathtt{aE}(\mathtt{act}, \overline{\mathtt{A}}, \mathtt{K}+1) \leftarrow \mathtt{act}(\overline{\mathtt{A}}, \mathtt{K});$$

to describe that effects of $\mathtt{act}$ can be realised by performing $\mathtt{act}$. Note that $\mathtt{act}$ has step variable $\mathtt{K}$ whereas $\mathtt{aE}$ has step variable $\mathtt{K}+1$, indicating the advance in time. Rule

$$\mathtt{hasAct}(\mathtt{K}) \leftarrow \mathtt{act}(\overline{\mathtt{A}}, \mathtt{K})$$

states that there is an action taking place at step $\mathtt{K}$ if there is an action $\mathtt{act}$ at $\mathtt{K}$. Rules

$$\mathtt{Ef_1}(\overline{\mathtt{E_1}}, \mathtt{K}) \leftarrow \mathtt{aE}(\mathtt{act}, \overline{\mathtt{A}}, \mathtt{K}); \ldots; \mathtt{Ef_m}(\overline{\mathtt{E_m}}, \mathtt{K}) \leftarrow \mathtt{aE}(\mathtt{act}, \overline{\mathtt{A}}, \mathtt{K})$$

describe each and every predicate listed as an effect of $\mathtt{act}$ can be derived from $\mathtt{aE}$, at the same time step as $\mathtt{aE}$. Rule

[7] When defining ABA frameworks, we omit to indicate the language component, as this can be easily inferred from the other components (being the set of all sentences occurring in rules, assumptions, and contraries). Also, we use rule schemata to simplify the notation. Each rule schema represents the set of grounded rules.

$$\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K}+1) \leftarrow \mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K}), \mathtt{hasAct}(\mathtt{K}), \mathtt{df}(\mathtt{FR}, \overline{\mathtt{A}}, \mathtt{K}+1)$$

describes that for every predicate FR describing a flexible relation, it is the case that if $\mathtt{FR}(\overline{\mathtt{A}})$ holds at step K, and there is some action taking place at step K, then it is assumed that, $\mathtt{FR}(\overline{\mathtt{A}})$ holds at step $\mathtt{K}+1$. The assumption $\mathtt{df}(\mathtt{FR}, \overline{\mathtt{A}}, \mathtt{K}+1)$ states that, by default, FR can be carried forward from K to $\mathtt{K}+1$. $\mathtt{hasAct}(\mathtt{K})$ is introduced in the body of the rule to enforce that at least one action has taken place during this step. Rule

$$\mathtt{RR}(\overline{\mathtt{A}}) \leftarrow$$

specifies that all rigid relations RR hold at all steps. Rules

$$\mathtt{P_1}(\overline{\mathtt{I_0}}, 0) \leftarrow; \ldots; \mathtt{P_n}(\overline{\mathtt{I_n}}, 0) \leftarrow.$$

specify that all predicates in the initial state s_0 hold at step 0.

Assumptions and contraries are such that:

- all actions are assumptions with contraries being either
  1. any other action at the same step or
  2. failure of meeting pre-conditions of the action.

  We enforce that any two different actions are in conflict at the same step; and if any pre-condition does not meet for an action, then the action cannot be performed.
- $\mathcal{C}(\mathtt{not\_PreC_i}(\overline{\mathtt{C_i}}, \mathtt{K})) = \{\mathtt{PreC_i}(\overline{\mathtt{C_i}}, \mathtt{K})\}$ specifies that the contrary of not meeting a pre-condition is meeting it.
- $\mathcal{C}(\mathtt{df}(\mathtt{FR}, \overline{\mathtt{A}}, \mathtt{K})) = \{\neg\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K})\}$ specifies that the assumption $\mathtt{df}(\mathtt{FR}, \overline{\mathtt{A}}, \mathtt{K})$ does not hold if it can be shown explicitly that $\mathtt{FR}(\overline{\mathtt{A}}, \mathtt{K})$ does not hold, at step K.

Example 5. (Example 2 continued.) Let the step bound $\mathbb{K} = 3$. The ABA framework corresponding to this planning problem is $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$, given as follows.[8]

- $\mathcal{R}$ is composed of the following rules.
  $\mathtt{goal} \leftarrow \mathtt{on}(\mathtt{d_1}, \mathtt{d_2}, 3), \mathtt{on}(\mathtt{d_2}, \mathtt{r_3}, 3);$
  $\mathtt{hA}(\mathtt{K}) \leftarrow \mathtt{mv}(\mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K});$ $\quad \mathtt{aE}(\mathtt{mv}, \mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K}) \leftarrow \mathtt{mv}(\mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K});$
  $\mathtt{cl}(\mathtt{A}, \mathtt{K}+1) \leftarrow \mathtt{aE}(\mathtt{mv}, \mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K});$ $\quad \neg\mathtt{cl}(\mathtt{B}, \mathtt{K}+1) \leftarrow \mathtt{aE}(\mathtt{mv}, \mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K});$
  $\mathtt{on}(\mathtt{D}, \mathtt{B}, \mathtt{K}+1) \leftarrow \mathtt{aE}(\mathtt{mv}, \mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K});$ $\neg\mathtt{on}(\mathtt{D}, \mathtt{A}, \mathtt{K}+1) \leftarrow \mathtt{aE}(\mathtt{mv}, \mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K});$
  $\mathtt{cl}(\mathtt{A}, \mathtt{K}+1) \leftarrow \mathtt{cl}(\mathtt{A}, \mathtt{K}), \mathtt{hA}(\mathtt{K}), \mathtt{df}(\mathtt{cl}, \mathtt{A}, \mathtt{K}+1);$ $\quad \mathtt{sm}(\mathtt{d_1}, \mathtt{d_2}) \leftarrow;$
  $\mathtt{on}(\mathtt{D}, \mathtt{A}, \mathtt{K}+1) \leftarrow \mathtt{on}(\mathtt{D}, \mathtt{A}, \mathtt{K}), \mathtt{hA}(\mathtt{K}), \mathtt{df}(\mathtt{on}, \mathtt{D}, \mathtt{F}, \mathtt{K}+1);$ $\mathtt{sm}(\mathtt{d_2}, \mathtt{r_3}) \leftarrow;$
  $\mathtt{sm}(\mathtt{d_1}, \mathtt{r_1}) \leftarrow; \mathtt{sm}(\mathtt{d_1}, \mathtt{r_2}) \leftarrow; \mathtt{sm}(\mathtt{d_1}, \mathtt{r_3}) \leftarrow; \mathtt{sm}(\mathtt{d_2}, \mathtt{r_1}) \leftarrow; \quad \mathtt{sm}(\mathtt{d_2}, \mathtt{r_2}) \leftarrow;$
  $\mathtt{cl}(\mathtt{r_2}, 0) \leftarrow; \ \mathtt{cl}(\mathtt{r_3}, 0) \leftarrow; \ \mathtt{cl}(\mathtt{d_1}, 0) \leftarrow; \ \mathtt{on}(\mathtt{d_1}, \mathtt{d_2}, 0) \leftarrow; \mathtt{on}(\mathtt{d_2}, \mathtt{r_1}, 0) \leftarrow;$
- $\mathcal{A}$ is composed of the following assumptions.
  $\mathtt{mv}(\mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K}), \mathtt{df}(\mathtt{on}, \mathtt{D}, \mathtt{F}, \mathtt{K}), \mathtt{df}(\mathtt{cl}, \mathtt{A}, \mathtt{K}), \mathtt{not\_on}(\mathtt{D}, \mathtt{F}, \mathtt{K}), \mathtt{not\_cl}(\mathtt{A}, \mathtt{K})$
- $\mathcal{C}$ is such that:
  $\mathcal{C}(\mathtt{mv}(\mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K})) = \{\mathtt{not\_sm}(\mathtt{D}, \mathtt{B}), \mathtt{not\_on}(\mathtt{D}, \mathtt{A}, \mathtt{K}), \mathtt{not\_cl}(\mathtt{D}, \mathtt{K}), \mathtt{not\_cl}(\mathtt{B}, \mathtt{K})\}$
  $\qquad \cup \{\mathtt{mv}(\mathtt{D}', \mathtt{A}', \mathtt{B}', \mathtt{K}) | \mathtt{mv}(\mathtt{D}', \mathtt{A}', \mathtt{B}', \mathtt{K}) \neq \mathtt{mv}(\mathtt{D}, \mathtt{A}, \mathtt{B}, \mathtt{K})\}$
  $\mathcal{C}(\mathtt{df}(\mathtt{on}, \mathtt{D}, \mathtt{F}, \mathtt{K})) = \{\neg\mathtt{on}(\mathtt{D}, \mathtt{F}, \mathtt{K})\}$ $\mathcal{C}(\mathtt{not\_on}(\mathtt{D}, \mathtt{F}, \mathtt{K})) = \{\mathtt{on}(\mathtt{D}, \mathtt{F}, \mathtt{K})\}$
  $\mathcal{C}(\mathtt{df}(\mathtt{cl}, \mathtt{A}, \mathtt{K})) = \{\neg\mathtt{cl}(\mathtt{A}, \mathtt{K})\}$ $\quad \mathcal{C}(\mathtt{not\_cl}(\mathtt{A}, \mathtt{K})) = \{\mathtt{cl}(\mathtt{A}, \mathtt{K})\}$

Now we are ready to present our main results on the correspondence between planning problems and ABA frameworks, with the next two theorems.

[8] hA is shorthand for hasAct.

Theorem 1. *Given a bounded planning problem $\mathcal{P}$ with a solution $S = (m_1, \ldots, m_n)$, let F be the ABA framework corresponding to $\mathcal{P}$. Then, there is a related admissible set of arguments RF in F with its topic sentence* goal *such that m_i in S are assumptions supporting arguments in RF.*

Proof. (Sketch.) We need to show that (1) RF is admissible, (2) all arguments in RF defend goal. Note that each m_i in S is of the form $\texttt{act}(\overline{\texttt{A}}, \texttt{i})$. To show (1), we start by observing that since S is a solution, there is an argument $A = \{\texttt{act}(\overline{\texttt{A}_\texttt{n}}, \texttt{n}), \ldots\} \vdash \texttt{goal}$ in F. Arguments attacking A are in the following three forms:

1. $\{\texttt{act}_\texttt{i}(\overline{\texttt{A}'_\texttt{i}}, \texttt{n})\} \vdash \texttt{act}_\texttt{i}(\overline{\texttt{A}'_\texttt{i}}, \texttt{n})$, targeting at the assumption $\texttt{act}(\overline{\texttt{A}}, \texttt{n})$, representing alternative moves one can make at step n,
2. $\{\texttt{not\_PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{n})\} \vdash \texttt{not\_PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{n})$, also targeting at $\texttt{act}(\overline{\texttt{A}}, \texttt{n})$, representing challenges to pre-conditions of the action at n,
3. $\Delta' \vdash \neg\texttt{FR}(\overline{\texttt{A}}, \texttt{n})$, targeting at assumptions $\texttt{df}(\texttt{FR}, \overline{\texttt{A}}, \texttt{n})$, representing challenges to flexible relations that they may not hold from one step to the next.

Attacking arguments in form (1) can be counterattacked by arguments $\{\texttt{act}(\overline{\texttt{A}}, \texttt{n})\} \vdash \texttt{act}(\overline{\texttt{A}}, \texttt{n})$ in RF. This can be read as, *although it is possible to make some other actions at step n, we can always choose to make action* act. Attacking arguments in form (2) can be counterattacked by arguments $\Delta^* \vdash \texttt{PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{n})$ in RF. Since S is a solution, all pre-conditions at each step must be met. This can be read as, *each action at step* n *meets all of its pre-conditions.* Attacking arguments in form (3), if they exist, indicate that there is some other plan S' such that $\neg\texttt{FR}(\overline{\texttt{A}}, \texttt{n})$ is in the goal state of S' and S' is not part of S. Since S is a solution, all pre-conditions in all of its actions must hold, for any S' composed by actions $\texttt{act}'(\overline{\texttt{B}}, \texttt{L}) \in \Delta'$ differ from the ones in S (up to step n), $\texttt{act}'(\overline{\texttt{B}}, \texttt{L})$ can be targeted by argument $\{\texttt{act}(\overline{\texttt{A}}, \texttt{L})\} \vdash \texttt{act}(\overline{\texttt{A}}, \texttt{L})$ in RF for $\texttt{act}(\overline{\texttt{A}}, \texttt{L})$ in S. This can be read as, *although there are some other plans S' such that S' invalidates some pre-condition of an action in S, S withstands such attacks as all of its actions meet their pre-conditions.* With this reasoning, we can see that all arguments in RF defend goal, thus meeting the second condition.

Theorem 2. *Given a bounded planning problem $\mathcal{P}$ with its corresponding ABA framework F, let RA be a related admissible set of arguments with topic sentence* goal*, $S = \{\texttt{act}(\overline{\texttt{A}}, \texttt{K}) | \texttt{act}(\overline{\texttt{A}}, \texttt{K})$ is the name of an action in $\mathbb{A}$ and $\texttt{act}(\overline{\texttt{A}}, \texttt{K})$ is an assumption of an argument in $RA\}$, then the sequence $(m_1, \ldots, m_n)$, for which $\{m_1, \ldots, m_n\} = S$ and $S(m_i) < S(m_j)$ if and only if $i < j$ is a solution to $\mathcal{P}$.*

Proof. (Sketch.) From Definition 3 we can see that RF is related admissible only when there is a sequence of actions taking the initial state to the goal state. Thus, all actions in assumptions defending goal consist a solution.

Theorem 1 and 2 sanction that bounded planning problems can be modelled with ABA frameworks such that solutions correspond to related admissible arguments. We illustrate these results with the following example.

Example 6. (Example 5 continued.) An argument for goal is $A = \Delta \vdash \texttt{goal}$ with $\Delta = \{\texttt{mv}(\texttt{d}_1, \texttt{r}_2, \texttt{d}_2, 2), \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1), \texttt{df}(\texttt{on}, \texttt{d}_2, \texttt{r}_3, 3)\}$ (see Fig. 2). Arguments targeting $\texttt{mv}(\texttt{d}_1, \texttt{r}_2, \texttt{d}_2, 2)$ include

$B = \{\texttt{mv}(\texttt{d}_1, \texttt{r}_2, \texttt{r}_1, 2)\} \vdash \texttt{mv}(\texttt{d}_1, \texttt{r}_2, \texttt{r}_1, 2)$ $C = \{\texttt{mv}(\texttt{d}_2, \texttt{r}_3, \texttt{r}_1, 2)\} \vdash \texttt{mv}(\texttt{d}_2, \texttt{r}_3, \texttt{r}_1, 2)$
$D = \{\texttt{not\_on}(\texttt{d}_1, \texttt{r}_2, 2)\} \vdash \texttt{not\_on}(\texttt{d}_1, \texttt{r}_2, 2)$ $E = \{\texttt{not\_cl}(\texttt{d}_1, 2)\} \vdash \texttt{not\_cl}(\texttt{d}_1, 2)$
$F = \{\texttt{not\_cl}(\texttt{d}_2, 2)\} \vdash \texttt{not\_cl}(\texttt{d}_2, 2)$ $G = \{\texttt{not\_sm}(\texttt{d}_1, \texttt{d}_2)\} \vdash \texttt{not\_sm}(\texttt{d}_1, \texttt{d}_2)$
Arguments B and C can be attacked by $H = \{\texttt{mv}(\texttt{d}_1, \texttt{r}_2, \texttt{d}_2, 2)\} \vdash \texttt{mv}(\texttt{d}_1, \texttt{r}_2, \texttt{d}_2, 2)$.
Arguments D can be attacked by
$I = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0), \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1), \texttt{df}(\texttt{on}, \texttt{d}_1, \texttt{r}_2, 2)\} \vdash \texttt{on}(\texttt{d}_1, \texttt{r}_2, 2)$ (Fig. 3)
Arguments E can be attacked by
$J = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0), \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1), \texttt{df}(\texttt{cl}, \texttt{d}_1, 2), \texttt{df}(\texttt{cl}, \texttt{d}_1, 1)\} \vdash \texttt{cl}(\texttt{d}_1, 2)$ (Fig. 4)
Arguments F can be attacked by
$K = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0), \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1), \texttt{df}(\texttt{cl}, \texttt{d}_2, 2)\} \vdash \texttt{cl}(\texttt{d}_2, 2)$ (Fig. 5)
Arguments G can be attacked by $L = \{\} \vdash \texttt{sm}(\texttt{d}_1, \texttt{d}_2)$.
We can see that arguments H, I, J, K and L all defend A. Arguments targeting $\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1)$ and $\texttt{df}(\texttt{on}, \texttt{d}_2, \texttt{r}_3, 3)$ can be counter-attacked similarly. H, J, K can also be defended with arguments in similar patterns. Overall, A is in a related admissible set with its topic `goal` and assumptions in Δ form a solution to the planning problem.

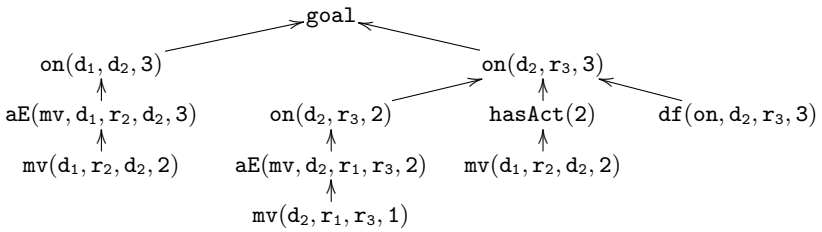

Fig. 2. An argument for `goal` in Example 6.

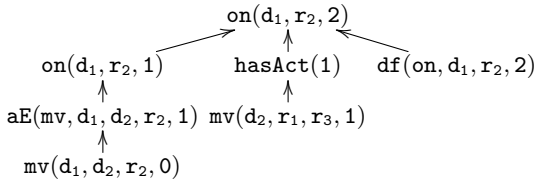

Fig. 3. An argument for $\texttt{on}(\texttt{d}_1, \texttt{r}_2, 2)$ in Example 6.

By Theorem 5 of [7], we know that assumptions in arguments held by proponent nodes of an admissible dispute tree are related admissible with the claim of the argument held by the root of the tree as the topic sentence. Thus, given Theorem 2, the following corollary holds.

Corollary 1. *Given a bounded planing problem $\mathcal{P} = \langle \mathbb{A}, s_0, s_g, \mathbb{F}, \mathbb{R}, \mathbb{K} \rangle$ with F the ABA framework corresponding to $\mathcal{P}$, if there is an admissible abstract dispute tree $\mathcal{T}^a$*

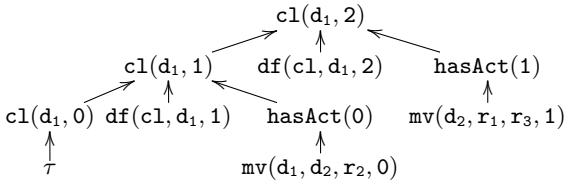

Fig. 4. An argument for $\texttt{cl}(\texttt{d}_1, 2)$ in Example 6.

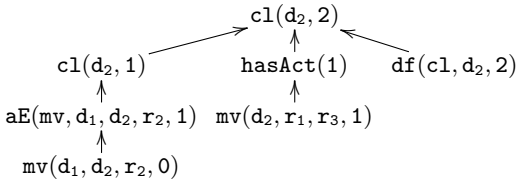

Fig. 5. An argument for $\texttt{cl}(\texttt{r}_2, 2)$ in Example 6.

for goal*, then* $\{\texttt{act}(\overline{\texttt{A}}, \texttt{K}) | \texttt{act}(\overline{\texttt{A}}, \texttt{K})$ *is a name of action in* $\mathbb{A}$ *and* $\mathbf{P} : \{\texttt{act}(\overline{\texttt{A}}, \texttt{K}), \ldots\} \vdash \_$ *is in* $\mathcal{T}^a\}$ *is a solution to* $\mathcal{P}$.[9]

Proof. Follows directly from Theorem 5 of [7] and Theorem 2.

Corollary 1 sanctions that abstract dispute trees can be used to compute solutions for bounded planning problems. This is a useful result as it allows us to use a semantics computation tool to compute plan solutions.

6 Extracting Explanations from ABA

In the previous section, we have shown how ABA can be used to compute solutions for bounded planning problems by identifying a related admissible set of arguments for the topic goal. Corollary 1 establishes the connection between solutions and abstract dispute trees. In this section, we focus on extracting explanations from dispute trees.

Proposition 1. *Given a bounded planning problem* $\mathcal{P}$ *with corresponding ABA framework* F*, let* E *be an explanation for* $\_ \vdash \texttt{goal}$ *in* F*. Then,* $S = s_g \cup \{s | \_ \vdash s \in E\}$ *contains an explanation for a plan* P *being a solution to* $\mathcal{P}$*, where* P *consists of actions represented by action assumptions supporting arguments in* E*.*

Proof. (Sketch.) By Theorem 2, and the definition of explanations for argument in ABA frameworks (see the Background section), P is a solution to $\mathcal{P}$. Since P is a solution, all pre-conditions of all actions in P must hold at their respective steps. By Definition 3, each action in the plan is mapped to an assumption with contraries being assumptions $\texttt{not\_PreC}_\texttt{i}$ for all of its pre-conditions. Since the contrary of $\texttt{not\_PreC}_\texttt{i}$ is $\texttt{PreC}_\texttt{i}$, there must be an admissible argument for each $\texttt{PreC}_\texttt{i}$. Thus, S by containing all goals in s_g and all pre-conditions of all actions, contains an explanation for P, by Definition 1.

[9] Throughout, _ stands for an anonymous variable.

Proposition 1 sanctions that, given a bounded planning problem $\mathcal{P}$ with its corresponding ABA framework F, by computing an explanation for the argument $\_ \vdash \texttt{goal}$, we not only compute a solution to $\mathcal{P}$, but also an explanation for this solution. We illustrate Proposition 1 with the following example.

Example 7. (Example 3 continued.) The ABA framework corresponding to this bounded planning problem (with $\mathbb{K} = 2, s_g = \{\texttt{on}(\texttt{d}_1, \texttt{r}_2, 2), \texttt{on}(\texttt{d}_2, \texttt{r}_3, 2)\}$) is the ABA framework shown in Example 5 with the rule

$$\texttt{goal} \leftarrow \texttt{on}(\texttt{d}_1, \texttt{d}_2, 3), \texttt{on}(\texttt{d}_2, \texttt{r}_3, 3)$$

replaced by

$$\texttt{goal} \leftarrow \texttt{on}(\texttt{d}_1, \texttt{r}_2, 2), \texttt{on}(\texttt{d}_2, \texttt{r}_3, 2)$$

and everything else unaltered. The argument A for goal is shown in Fig. 6. Arguments attacking A are summarised in Table 1. Argument B targets at assumption $\texttt{df}(\texttt{on}, \texttt{d}_1, \texttt{r}_2, 2)$. Arguments $C1$-$C5$ target at assumption $\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0)$. Arguments $D1$-$D5$ target at assumption $\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1)$. Note that $B, C1$ and $D1$ represent sets of arguments with A unified to different r_is and d_js.

Arguments attacking $B, C1$-$C5$ and $D1$-$D5$ are shown in Table 2. Here, X' attack X, for $X = B$, $C1$-$C5$, $D1$-$D5$. Arguments attacking B' and $D1'$ are $D1$-$D5$. Arguments attacking $C1'$ are $C1$-$C5$. Arguments attacking $D2'$ are: $C1$ - $C5$ and $E = \{\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{A}, 0)\} \vdash \neg\texttt{on}(\texttt{d}_2, \texttt{r}_1, 1)$. Arguments attacking $D3'$ are: $C1$ - $C5$ and $F = \{\texttt{mv}(\texttt{A}, \texttt{B}, \texttt{d}_2, 0)\} \vdash \neg\texttt{cl}(\texttt{d}_2, 1)$. Arguments attacking $D4'$ are: $C1$ - $C5$ and $G = \{\texttt{mv}(\texttt{A}, \texttt{B}, \texttt{d}_3, 0)\} \vdash \neg\texttt{cl}(\texttt{r}_3, 1)$. Argument $C1'$ attacks E, F and G. In summary, we can see that the related admissible set of arguments, which is the defence set of the tree, together with the goal set, contains explanations for the plan being a solution.

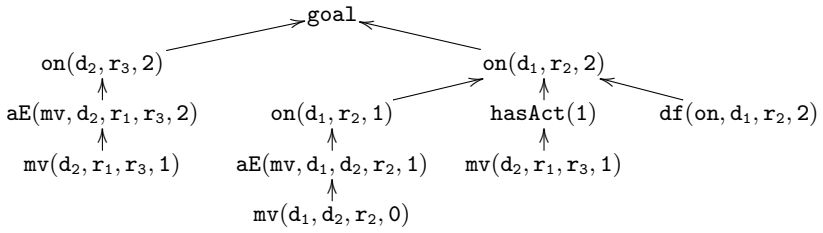

Fig. 6. An argument for goal in Example 7.

Proposition 2. *Given a bounded planning problem $\mathcal{P}$ with corresponding ABA framework F, let $P = (m_1, \ldots, m_n)$ be an invalid plan. If there is a non-admissible dispute tree $\mathcal{T}^a$ for $\_ \vdash \texttt{goal}$ such that $m_1, \ldots, m_n$ support arguments held by $\mathbf{P}$ nodes in $\mathcal{T}^a$. Then, if a node $N = \mathbf{P} : \{\texttt{m}_\texttt{i}, \ldots\} \vdash \_$ is defeated and $A = \{\texttt{not\_PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{K})\} \vdash \texttt{not\_PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{K})$ held by N's winning attacker, then $\texttt{PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{K})$ is in an explanation for the invalidity of P.*

Proof. (Sketch.) By Definition 2, to show that $\texttt{PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{K})$ is in an explanation for the invalidity of P is to show that $\texttt{PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{K})$ is not held at K. This is achieved by

Table 1: Arguments attacking A in Example 7.

$B = \{\texttt{mv}(\texttt{d}_1, \texttt{r}_2, \texttt{A}, 1)\} \vdash \neg\texttt{on}(\texttt{d}_1, \texttt{r}_2, 2)$

$C1 = \{\texttt{mv}(\texttt{A}, \texttt{B}, \texttt{C}, 0)\} \vdash \texttt{mv}(\texttt{A}, \texttt{B}, \texttt{C}, 0)$ $C2 = \{\texttt{not\_on}(\texttt{d}_1, \texttt{d}_2, 0)\} \vdash \texttt{not\_on}(\texttt{d}_1, \texttt{d}_2, 0)$

$C3 = \{\texttt{not\_cl}(\texttt{d}_1, 0)\} \vdash \texttt{not\_cl}(\texttt{d}_1, 0)$ $C4 = \{\texttt{not\_cl}(\texttt{r}_2, 0)\} \vdash \texttt{not\_cl}(\texttt{r}_2, 0)$

$C5 = \{\texttt{not\_sm}(\texttt{d}_1, \texttt{r}_2)\} \vdash \texttt{not\_sm}(\texttt{d}_1, \texttt{r}_2)$ $D1 = \{\texttt{mv}(\texttt{A}, \texttt{B}, \texttt{C}, 1)\} \vdash \texttt{mv}(\texttt{A}, \texttt{B}, \texttt{C}, 1)$

$D2 = \{\texttt{not\_on}(\texttt{d}_2, \texttt{r}_1, 1)\} \vdash \texttt{not\_on}(\texttt{d}_2, \texttt{r}_1, 1)$ $D3 = \{\texttt{not\_cl}(\texttt{d}_2, 1)\} \vdash \texttt{not\_cl}(\texttt{d}_2, 1)$

$D4 = \{\texttt{not\_cl}(\texttt{r}_3, 1)\} \vdash \texttt{not\_cl}(\texttt{r}_3, 1)$ $D5 = \{\texttt{not\_sm}(\texttt{d}_1, \texttt{r}_3)\} \vdash \texttt{not\_sm}(\texttt{d}_1, \texttt{r}_3)$

Table 2: Arguments attacking $B, C1$-$C5$ and $D1$-$D5$ in Example 7.

$B' = \{\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1)\} \vdash \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1)$ $C1' = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0)\} \vdash \texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0)$

$C2' = \{\} \vdash \texttt{on}(\texttt{d}_1, \texttt{d}_2, 0)$ $C3' = \{\} \vdash \texttt{cl}(\texttt{d}_1, 0)$

$C4' = \{\} \vdash \texttt{cl}(\texttt{r}_2, 0)$ $C5' = \{\} \vdash \texttt{sm}(\texttt{d}_1, \texttt{r}_2)$

$D1' = \{\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1)\} \vdash \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_3, 1)$

$D2' = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0), \texttt{df}(\texttt{on}, \texttt{d}_2, \texttt{r}_1, 1)\} \vdash \texttt{on}(\texttt{d}_2, \texttt{r}_1, 1)$

$D3' = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0), \texttt{df}(\texttt{cl}, \texttt{d}_2, 1)\} \vdash \texttt{cl}(\texttt{d}_2, 1)$

$D4' = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_2, 0), \texttt{df}(\texttt{cl}, \texttt{r}_3, 1)\} \vdash \texttt{cl}(\texttt{r}_3, 1)$ $D5' = \{\} \vdash \texttt{sm}(\texttt{d}_2, \texttt{r}_3)$

showing that the argument A held in $\mathcal{T}^a$. Since A is held by a winning attacker of N, in $\mathcal{T}^a$, meaning that it cannot be disapproved, $\texttt{PreC}_\texttt{i}(\overline{\texttt{C}_\texttt{i}}, \texttt{K})$ cannot be held. Thus it is in an explanation for the invalidity of P.

Proposition 2 sanctions that to identify explanations for invalid actions, we can look at dispute trees and find non-admissible sub-trees and their successful attackers. We illustrate Proposition 2 with the following example.

Example 8. (Example 4 continued.) Let the goal state $s_g = \{\texttt{on}(\texttt{d}_1, \texttt{r}_3, 2), \texttt{on}(\texttt{d}_2, \texttt{d}_1, 2)\}$. An argument for goal is $A = \Delta \vdash \texttt{goal}$ with $\Delta = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0), \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1), \texttt{df}(\texttt{on}, \texttt{d}_1, \texttt{r}_3, 2)\}$. Arguments targeting at $\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1)$ include the following:

$B = \{\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_2, 1)\} \vdash \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{r}_2, 1)$,

$C = \{\texttt{not\_on}(\texttt{d}_2, \texttt{r}_1, 1)\} \vdash \texttt{not\_on}(\texttt{d}_2, \texttt{r}_1, 1)$,

$D = \{\texttt{not\_cl}(\texttt{d}_1, 1)\} \vdash \texttt{not\_cl}(\texttt{d}_1, 1)$,

$E = \{\texttt{not\_cl}(\texttt{d}_2, 1)\} \vdash \texttt{not\_cl}(\texttt{d}_2, 1)$,

$F = \{\texttt{not\_sm}(\texttt{d}_2, \texttt{d}_1)\} \vdash \texttt{not\_sm}(\texttt{d}_2, \texttt{d}_1)$.

$G = \{\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1)\} \vdash \{\texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1)\}$ attacks B.

$H = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0), \texttt{df}(\texttt{on}, \texttt{d}_2, \texttt{r}_1, 1)\} \vdash \texttt{on}(\texttt{d}_2, \texttt{r}_1, 1)$ attacks C.

$I = \{\texttt{df}(\texttt{cl}, \texttt{d}_1, 1), \texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0)\} \vdash \texttt{cl}(\texttt{d}_1, 1)$ attacks D.

$J = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0)\} \vdash \texttt{cl}(\texttt{d}_2, 1)$ attacks E. However, no argument can attack F as there is no argument for $\texttt{sm}(\texttt{d}_2, \texttt{d}_1)$. Thus, $\mathbf{O} : F$ is a winning attacker and $\mathbf{P} : A$ is defeated. A fraction of the abstract dispute tree for A is shown in Fig. 7. Thus, we conclude that the unmet pre-condition $\texttt{sm}(\texttt{d}_2, \texttt{d}_1)$ is in an explanation for the invalidity of the plan.

For simplicity, we only present results for plan invalidity due to unmet pre-conditions. In general, however, by Definition 2, a plan is invalid if there are multiple actions take place at the same time. It is easy to see that such invalidity can be easily captured in dispute trees as the contrary of an assumption representing an action includes all other

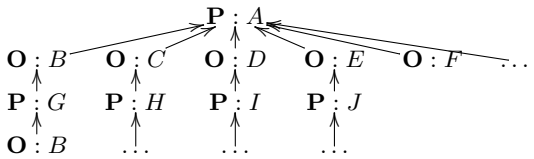

Fig. 7. An illustration of a fraction of the non-admissible dispute tree for $\Delta \vdash \texttt{goal}$ with $\Delta = \{\texttt{mv}(\texttt{d}_1, \texttt{d}_2, \texttt{r}_3, 0), \texttt{mv}(\texttt{d}_2, \texttt{r}_1, \texttt{d}_1, 1), \texttt{df}(\texttt{on}, \texttt{d}_1, \texttt{r}_3, 2)\}$ in Example 8. Since $\mathbf{O} : F$ has no $\mathbf{P}$ child, $\mathbf{O} : F$ is a winning attacker.

actions taking place at the same step. Therefore, if a defeated node labelled by an argument supported by an action assumption and the winning attacker holds an argument with an action as its claim, then the action in the winning attacker is in an explanation.

7 Related Work

Very recently, [9] presented a study on "Explainable Planing". Not connected to argumentation, the authors proposed six questions to be answered by "explainable planners". That paper focuses on high level discussion with no theoretical result. The two forms of explanations presented in this work can be viewed as (at least partial) answers to three of their questions: **Q1:** *Why did you do that?* (because such moves are valid), **Q2** *Why didn't you do something else?* (because "something else" is not valid) and **Q4:** *Why can't you do that?* (because certain precondition does not meet), as our explanations justify actions in plans and identify invalid ones.

Argumentation has seen its use in planning since 1980s (see e.g. [11, 13]), in the context of urban planning. Formal arguments are constructed to evaluate the pros and cons of urban plans. Our work differs from theirs as we view plans as sequences of actions and argumentation is used to generate these sequences.

Connection between planning and defeasible argumentation has been made in [10] with an application in [8]. Their works are based on DeLP where arguments are in the level of actions and a tailored algorithm has been developed for searching the suitable plans. Our work differs from theirs as we use ABA arguments to represent plans with existing argumentation semantics computing techniques so that explanations can be obtained along the course.

Conflict resolution in goal selection and planning has been studied by researchers in argumentation [1, 12]. Our work differs from theirs as, in addition to using entirely different argumentation languages, we use argumentation as a modelling tool for solving planning problems and derive explanations from our solutions; whereas they use argumentation to model agent actions and desires so that more suitable actions can be selected.

In multi-agent negotiation, there are works on connecting argumentation with the classic planners [15, 18]. There, argumentation-based negotiation is viewed as a planning problem with negotiation utterances being agents' actions. Under such modelling, negotiation is then solved with existing planner, e.g., HTN. Similarly, argumentation-based persuasion can be viewed as a planning problem and solved with existing plan-

ning techniques [3]. Our work is orthogonal to those as we use argumentation to plan instead of using planner to argue.

Agent planning has been studied in [2]. There, agents' beliefs and actions are analysed in a single argumentation framework, such that plans satisfying agents' desires can be directly generated. Our work differs from theirs as, instead of studying BDI agent planning in a specifically defined language, we present a generic representation for planning problems represented in STRIPS. Moreover, by explicity representing planning steps, our approach reasons with temporal information, which is not supported in [2]. Thus, a better generalisation and improved applicability have been achieved.

8 Conclusion

Empowering AI techniques with the ability of generating explanations is central to bringing trust to autonomous systems. In this paper, we have looked at how to use ABA to model planning problems and to generate explanations. The aim is to develop planning techniques which not only produce solutions, but also generate explanations for solutions (and non-solutions). Taking a generic planning problem represented in a standard STRIPS-like language, our model generates an "ABA counterpart" of the problem. The correspondence is realised such that plan solutions correspond to Related Admissible arguments with the topic being the goal state. The proposed plan construction method is both sound and complete in the sense that a solution exists if and only if the corresponding Related Admissible arguments exist.

To generate planning explanations, we again rely on the Related Admissible semantics and its computation means, dispute trees. We observe that a plan is a solution if and only if all goals are met at the end of the plan and there is no unmet pre-condition in any action in the plan. Related Admissible set of arguments computed with dispute trees contain justifications for all actions meeting their pre-conditions. Moreover, by looking at "defeated nodes" in non-admissible dispute trees, we identify unmet pre-conditions, which explain why some plans are not solutions.

In future, we will further explore argumentation-based explanations in planning. Namely, we would like to study explanations for questions such as "Why plan A is better than plan B?", "Why certain goal can never be reached by any plan?". We will also study argumentation-based dialectical explanation, which can be used in multi-agent planning. Moreover, we will look at mapping other planning languages with argumentation, e.g. support partial order planning and conditional-effects of actions.

References

1. L. Amgoud and C. Cayrol. On the use of an ATMS for handling conflicting desires. In *Proc. of KR*, pages 194–202, 2004.
2. L. Amgoud, C. Devred, and M. Lagasquie-Schiex. Generating possible intentions with constrained argumentation systems. *IJAR*, 52(9):1363–1391, 2011.
3. E. Black, A. J. Coles, and C. Hampson. Planning for persuasion. In *Proc. of AAMAS*, pages 933–942, 2017.
4. K. Čyras, X. Fan, C. Schulz, and F. Toni. Assumption-based argumentation: Disputes, explanations, preferences. *IfCoLog JLTA*, 4(8), 2017.

5. D. Doran, S. Schulz, and T. R. Besold. What does explainable AI really mean? A new conceptualization of perspectives. *CoRR*, abs/1710.00794, 2017.
6. P.M. Dung, R.A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *AIJ*, 170:114–159, 2006.
7. X. Fan and F. Toni. On computing explanations in argumentation. In *Proc. of AAAI*, pages 1496–1502, 2015.
8. S. P. Ferrando and E. Onaindia. Defeasible argumentation for multi-agent planning in ambient intelligence applications. In *Proc. of AAMAS*, pages 509–516, Richland, SC, 2012. IFAAMS.
9. M. Fox, D. Long, and D. Magazzeni. Explainable planning. *CoRR*, abs/1709.10256, 2017.
10. D. R. García, A. J. García, and G. R. Simari. Defeasible reasoning and partial order planning. In *Proc. of FoIKS*, pages 311–328, Berlin, Heidelberg, 2008. Springer-Verlag.
11. H A Goldstein. Planning as argumentation. *Environment and Planning B: Planning and Design*, 11(3):297–312, 1984.
12. J. Hulstijn and L. W. N. van der Torre. Combining goal generation and planning in an argumentation framework. In *Proc. NMR*, pages 212–218, 2004.
13. K Lapintie. Analysing and evaluating argumentation in planning. *Environment and Planning B: Planning and Design*, 25(2):187–204, 1998.
14. S. Modgil, F. Toni, F. Bex, I. Bratko, C. Chesñevar, W. Dvořák, M Falappa, X. Fan, S. Gaggl, A. García, M. González, T. Gordon, J. Leite, M. Možina, C. Reed, G. Simari, S. Szeider, P. Torroni, and S. Woltran. The added value of argumentation. In *Agreement Technologies*, volume 8, pages 357–403. Springer, 2013.
15. A. Monteserin and A. Amandi. Argumentation-based negotiation planning for autonomous agents. *Decision Support System*, 51(3):532–548, June 2011.
16. D. Nau, M. Ghallab, and P. Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
17. N. J. Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1980.
18. A. R. Panisson, G. Farias, A. Fraitas, F. Meneguzzi, R. Vieira, and R. H. Bordini. Planning interactions for agents in argumentation-based negotiation. In *Proc. of ArgMAS*, 2014.
19. M. Vallati, L. Chrpa, M. Grzes, T. L. McCluskey, M. Roberts, and S. Sanner. The 2014 international planning competition: Progress and trends. *AI Magazine*, 36(3):90–98, 2015.
20. S. Wachter, B. Mittelstadt, and L. Floridi. Transparent, explainable, and accountable ai for robotics. *Science Robotics*, 2(6), 2017.