A Temporal Planning Example with Assumption-based Argumentation

Xiuyi Fan

Swansea University, Swansea, United Kingdom,
xiuyi.fan@swansea.ac.uk

Abstract. Agent planning has attracted much research attention in recent years. In argumentation, agent planning has been studied by several researchers with significant contributions made in modelling agent goals, desires and actions. However, there is little work that connects argumentation semantics, plan construction and temporal information in a unified framework. In this work, we use a version of the classic *blocks world* planning problem as our case study and demonstrate how Assumption-based Argumentation can be used to tackle planning problems with explicit time step information. In our approach, the process of plan construction is equated to constructing acceptable arguments (with respect to an argumentation semantics) with temporal aspects taken into consideration.

1 Introduction

Agent planning has been studied with argumentation-based approaches by several researchers. Notably, Amgoud and her colleagues have studied joining *deliberation* and *means-ends reasoning* in a single unified argumentation system such that "[the] system combines option generation and checking the feasibility of options" [1, 2]. In their work, argumentation has been used to identify agent intentions in a way that the resulting intentions satisfy the argumentation rationality postulates [2]. However, in their work, temporal reasoning has not been considered as agent goals are fulfilled by *sets* of actions instantaneously rather than through *a sequence* of actions over a course of plan execution. On the other hand, García et al. [4] have studied incorporating defeasible information in agent planning, as originally proposed by Pollock [6]. In García's work, argumentation has been introduced to model defeasibility in planning [4]. Their work is more inline with classic planning approaches (see e.g. [5]) in that their plans are sequences of actions with effects. Although temporal information has been considered in [4] with arguments modelling defeasibility, they have used a specifically designed search process to identify suitable plans. Thus their plans are not confirmed to argumentation rationality postulates in the same way as [2].

In this work, we study using Assumption-based Argumentation (ABA) [3] to solve planning problems. In the same spirit as [4], we consider plans consist sequences of actions and performing an action results changes to the "world state". Thus, the execution of a plan transfers the world from some "initial situation" to a "final situation". The designed transformation is carried out argumentatively so argumentation semantics can be used to validate plans. Unlike [4], where a dedicated algorithm was introduced to search

for plan solutions, we equate plan solution construction with acceptable argument computation. In our work, actions are modelled with assumptions and the contrary of an assumption describe conditions for which an action cannot be performed; the efforts of an action describe changes to the world state, updating available actions. Overall, our work can be viewed as an illustration of an ABA instantiation of Pollock's idea on defeasible planning: "It is argued that the planning must instead be done defeasibly, making the default assumption that there are no threats and then modifying plans as threats are discovered." [6]

2 Background

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

- $\langle \mathcal{L}, \mathcal{R} \rangle$ 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 a rule $\rho = s_0 \leftarrow s_1, \ldots, s_m$, s_0 is referred to as the *head* and $s_1, \ldots, s_m$ as the *body*. A rule with an empty body is referred to as a *fact*.

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 $\Delta_2 \vdash s_2$ iff 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* iff 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* iff it belongs to an admissible set of arguments.

3 Planning in Blocks World

A blocks world (Fig. 1) contains a set of "blocks" of different sizes and a set of "locations" where each block is at some location, and each location could have a "block pile" such that if two blocks r, r' are in a pile then r is placed higher than r', iff r is smaller than r'. The initial and final situations are two placements of blocks. Two types of actions are possible to a block, (1) moving it from one location to another while satisfying the "smaller-block-placed-higher" constraint and (2) no-operation, i.e., not to move it. A plan is a sequence of actions which transfers the initial situation to the final one. Formally, we use the following four definitions.

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

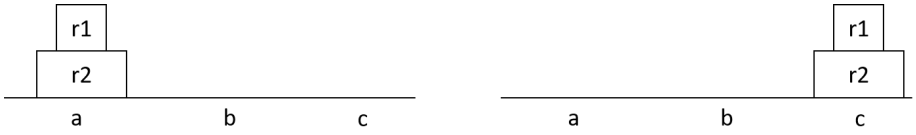

Fig. 1. A blocks world with two blocks $r1$ and $r2$ and three locations a, b and c. The initial situation is shown on the left-hand side and the final situation is shown on the right-hand side.

Definition 1. *A* blocks world *is a tuple* $\langle \mathtt{R}, \mathtt{L}, < \rangle$ *in which* $\mathtt{R}$ *is a set of* blocks*,* $\mathtt{L}$ *is a set of* locations *and* $< \subseteq \mathtt{R} \times \mathtt{R}$ *is a total order such that for* $r_1, r_2 \in \mathtt{R}$*,* $r_1 < r_2$ *iff* r_1 *is smaller than* r_2*.*

Definition 2. *A* blocks world planning problem *is a tuple* $\langle W, \mathtt{T}, S_0, S_n \rangle$ *where*

- $W = \langle \mathtt{R}, \mathtt{L}, < \rangle$ *is a blocks world,*
- $\mathtt{T} = \langle t_0, \ldots, t_n \rangle$ *is a time step sequence,*
- *a* situation S_i *(for a time step* t_i *in* $\mathtt{T}$*) is a set* $\{at(r_1, l_1, t_i), \ldots, at(r_n, l_m, t_i)\}$ *specifying the location* $l_j \in \mathtt{L}$ *for each block* $r_k \in \mathtt{R}$*. For a situation* S_i*, there is no* $r_k \in \mathtt{R}$ *such that* r_k *is not specified in* S_i*. Moreover,*
  - $S_0 = \{at(r_1, l_1, t_0), \ldots, at(r_n, l_m, t_0)\}$ *is the* initial situation*, and*
  - $S_n = \{at(r_1, l'_1, t_n), \ldots, at(r_n, l'_m, t_n)\}$ *is the* final situation*.*

Definition 3. *In a blocks world* $\langle \mathtt{R}, \mathtt{L}, < \rangle$*, the two* actions *are: (1)* $move(X, L, L', T)$*, move* $X \in \mathtt{R}$ *from* $L \in \mathtt{L}$ *to* $L' \in \mathtt{L}$ *at time step* T*, and (2)* $noOp(X, L, T)$*, do nothing to* $X \in \mathtt{R}$ *sitting at* $L \in \mathtt{L}$ *at time* T*.*

Given $S_i = \{at(r_1, l_1, t_i), \ldots, at(r_n, l_i, t_i)\}$*, then* apply $move(r_i, l, l', t_i)$ *to* S_i *yield* $(S_i \setminus \{at(r_i, l, t_i)\}) \cup \{at(r_i, l', t_{i+1})\}$*;* apply $noOp(r_i, l, t_i)$ *to* S_i *yield* $(S_i \setminus \{at(r_i, l, t_i)\}) \cup \{at(r_i, l, t_{i+1})\}$*.*

$move(X, L, L', T)$ *is* valid *iff all of the following conditions hold (at time* T*):*

(C1) X *sits at* L*;*
(C2) there is no $X' \in \mathtt{R}$ *such that* X' *is at* L *and* $X' < X$*;*
(C3) if there is a block at L'*, then let* $X' \in \mathtt{R}$ *be the top block at* L'*,* $X < X'$*;*
(C4) X *is not being moved to a different location* $L' \in \mathtt{L}$*; and*
(C5) there is no other block $X' \in \mathtt{R}$ *being moved to* L*.*

Definition 4. *A* plan *for* $\langle W, \mathtt{T}, S_0, S_n \rangle$ *is a set*

$$\{P(r_1, t_0) \ldots P(r_m, t_0)\} \cup \ldots \cup \{P(r_1, t_n) \ldots P(r_m, t_n)\}$$

where each $P(r_i, t_j)$ *is either* $move(r_i, l, l', t_j)$ *or* $noOp(r_i, l, t_j)$ *such that* S_1 *is obtained by applying all* $P(r, 0)$ *to* S_0*;* S_{i+1} *is obtained by applying all* $P(r, t_i)$ *to* S_i*; and* S_n *is obtained by applying all* $P(r, t_{n-1})$ *to* S_{n-1}*.*

A plan P *is* valid *iff all actions in* P *are valid; otherwise,* P *is* invalid*.*

We illustrate Definition 2-4 with the following example.

Example 1. Given the blocks world shown in Figure 1, we have blocks $\mathtt{R} = \{r_1, r_2\}$, locations $\mathtt{L} = \{a, b, c\}$ and time steps $\mathtt{T} = \langle t_0, t_1, t_2, t_3 \rangle$. Moreover, r_1 is smaller than r_2. The initial situation is $\{at(r_1, a, t_0), at(r_2, a, t_0)\}$ and the final situation is $\{at(r_1, c, t_3), at(r_2, c, t_3)\}$. It is easy to see that $\{move(r_1, a, b, t_0)$, $noOp(r_2, a, t_0)$, $noOp(r_1, b, t_1)$, $move(r_2, a, c, t_1)$, $move(r_1, b, c, t_2)$, $noOp(r_2, c, t_2)\}$ is a plan.

4 Planning in Blocks World with ABA

We take a two-step approach to model blocks world planning with ABA. Firstly, we define the *core framework* wrt a blocks world; then, for any given specific problem with a given time step sequence and initial, final situations, we define an *instantiated framework* (extending the core framework) to generate specific plans. Formally,

Definition 5. *Given a blocks world $W = \langle \mathtt{R}, \mathtt{L}, < \rangle$, the core framework corresponding to W is an ABA framework $F_0 = \langle \mathcal{L}_0, \mathcal{R}_0, \mathcal{A}_0, \mathcal{C}_0 \rangle$ such that:*[2]

- *$\mathcal{R}_0$ contains the following rules and nothing else.*
  $above(X, X', L, T) \leftarrow at(X, L, T), at(X', L, T), smaller(X, X')$ *(1)*
  $at(X, L, T) \leftarrow at(X, L, T^-), noOp(X, L, T^-), succ(T, T^-)$ *(2)*
  $at(X, L, T) \leftarrow at(X, L', T^-), move(X, L', L, T^-), succ(T, T^-)$ *(3)*
  $occupied(X, L, T) \leftarrow at(X', L, T), smaller(X', X)$ *(4)*
  $smaller(X, X') \leftarrow$ *(5)*
- *$\mathcal{A}_0$ contains the following assumptions and nothing else.*
  $move(X, L', L, T) \qquad noOp(X, L, T) \qquad notAt(X, L, T)$
- *$\mathcal{C}_0$ is such that:*
  $\mathcal{C}(move(X, L', L, T)) = \{noOp(X, L, T), move(X, L', L'', T), move(X', L'', L, T), above(X', X, L, T), occupied(X, L, T), notAt(X, L', T)\}$
  $\mathcal{C}(noOp(X, L, T)) = \{move(X, L, L', T)\}$
  $\mathcal{C}(notAt(X, L, T)) = \{at(X, L, T)\}$

Rule 1 states that a block X is above another block X' at time T if both X and X' are at the same location L at time T and X is smaller than X'. Rule 2 states that if a block X is at location L at t_i and it is not moved at t_i, then X is at L at time t_{i+1}. Rule 3 states that, at t_i, if a block X is at location L' and X is moved from L' to L, then X is at L at t_{i+1}. Rule 4 states that a location L is occupied wrt to a block X if there is another block X' at L such that X' is smaller than X. Rule 5 states that X is smaller than X' for all $X < X'$.

Assumptions $\mathcal{A}_0$ and their contraries $\mathcal{C}_0$ can be read as:

1. we move a block X from L' to L at time T unless (a) we do not move X, or (b) we move it to a different L'', or (c) some other block $X' \neq X$ is moved to L, or (d) some other block X' is on top of X, or (e) X is no smaller than the top of pile block at L, or (f) X is not at L'.
2. we do not move a block X unless we move it;
3. a block X is not at a location L unless it is at L.

Within a blocks world, a planning problem can be modelled with an instantiated framework, defined as follows.

[2] We use rule and assumption schemata to simplify our notations. Specifically, in each of the rules, assumptions and contraries, we have $X, X' \in \mathtt{R}$, $X \neq X'$, $L, L', L'' \in \mathtt{L}$, $L \neq L'$, $L \neq L''$, $L' \neq L''$ and T, T^- in some time step sequence. In Rule 5, we also enforce that $X < X'$.

Table 1: Arguments attacking A in Example 2.

Arguments targeting at $move(r_1, a, b, t_0)$:	
$B_1 = \{m(r_1, a, c, t_0)\} \vdash m(r_1, a, c, t_0)$,	$B_2 = \{m(r_2, a, b, t_0)\} \vdash m(r_2, a, b, t_0)$,
$B_3 = \{m(r_2, c, b, t_0)\} \vdash m(r_2, c, b, t_0)$,	$B_4 = \{n(r_1, a, t_0)\} \vdash n(r_1, a, t_0)$,
$B_5 = \{nA(r_1, a, t_0)\} \vdash nA(r_1, a, t_0)$.	
Arguments targeting at $n(r_2, a, t_0)$:	
$C_1 = \{m(r_2, a, b, t_0)\} \vdash m(r_2, a, b, t_0)$,	$C_2 = \{m(r_2, a, c, t_0)\} \vdash m(r_2, a, c, t_0)$.
Arguments targeting at $n(r_1, b, t_1)$:	
$D_1 = \{m(r_1, b, a, t_1)\} \vdash m(r_2, b, a, t_1)$,	$D_2 = \{m(r_1, b, c, t_1)\} \vdash m(r_2, b, c, t_1)$.
Arguments targeting at $m(r_2, a, c, t_1)$:	
$E_1 = \{m(r_2, a, b, t_1)\} \vdash m(r_2, a, b, t_1)$,	$E_2 = \{m(r_1, a, c, t_1)\} \vdash m(r_1, a, c, t_1)$,
$E_3 = \{m(r_1, b, c, t_1)\} \vdash m(r_1, b, c, t_1)$,	$E_4 = \{n(r_2, a, t_1)\} \vdash n(r_2, a, t_1)$,
$E_5 = \{nA(r_2, a, t_1)\} \vdash nA(r_2, a, t_1)$,	$E_7 = \{m(r_1, a, c, t_0)\} \vdash o(r_2, c, t_1)$,
$E_6 = \{n(r_1, a, t_0), n(r_2, a, t_0)\} \vdash above(r_1, r_2, a, t_1)$.	
Arguments targeting at $m(r_1, b, c, t_2)$:	
$F_1 = \{m(r_1, b, a, t_2)\} \vdash m(r_1, b, a, t_2)$,	$F_2 = \{m(r_2, a, c, t_2)\} \vdash m(r_2, a, c, t_2)$,
$F_3 = \{m(r_2, b, c, t_2)\} \vdash m(r_2, b, c, t_2)$,	$F_4 = \{n(r_1, b, t_2)\} \vdash n(r_1, b, t_2)$,
$F_5 = \{nA(r_1, b, t_2)\} \vdash nA(r_1, b, t_2)$.	
Arguments targeting at $n(r_2, c, t_2)$:	
$G_1 = \{m(r_2, c, a, t_2)\} \vdash m(r_2, c, a, t_2)$,	$G_2 = \{m(r_2, c, b, t_2)\} \vdash m(r_2, c, b, t_2)$.

Definition 6. *For a planning problem $\Pi = \langle W, \mathtt{T}, S_0, S_n\rangle$, let $\langle \mathcal{L}_0, \mathcal{R}_0, \mathcal{A}_0, \mathcal{C}_0\rangle$ be the core framework for W, then the instantiated framework corresponding to Π is an ABA framework $F_I = \langle \mathcal{L}_I, \mathcal{R}_I, \mathcal{A}_I, \mathcal{C}_I\rangle$ such that:*

- *$\mathcal{R}_I$ is $\mathcal{R}_0$ with the following additional rules:*
  $goal \leftarrow s_1, \ldots, s_m$, *for* $\{s_1, \ldots, s_m\} = S_n$, *(1)*
  $succ(T, T^-) \leftarrow$, *for all* $T, T^- \in \mathtt{T}$ *such that* T *is the successor of* T^-, *(2)*
  $s \leftarrow$ *for each* $s \in S_0$; *(3)*
- *$\mathcal{A}_I = \mathcal{A}_0$, and for each $\alpha \in \mathcal{A}_I$, $\mathcal{C}_I(\alpha) = \mathcal{C}_0(\alpha)$.*

The core framework corresponding to a blocks world W capturing generic information about W. The instantiated framework encodes information that is specific to a planning problem. Namely, $\mathcal{R}_I$ contains all rules in $\mathcal{R}_0$ and with an addition rule to describe what is to be achieved in the final situation (Rule 1), facts to describe the time step sequence (Rule 2), and facts to describe the initial situation (Rule 3). We illustrate Definition 5 and 6 with the following example.

Example 2. (Example 1 continued.) As given in Definition 6, we introduce rules
$goal \leftarrow at(r_1, c, t_3), at(r_2, c, t_3)$
$at(r_1, a, t_3) \leftarrow at(r_2, a, t_3) \leftarrow smaller(r_1, r_2) \leftarrow$
in the instantiated framework. An admissible argument for $goal$, $A = \Delta \vdash goal$, is shown in Figure 2 with $\Delta = \{move(r_1, a, b, t_0)$, $noOp(r_2, a, t_0)$, $noOp(r_1, b, t_1)$, $move(r_2, a, c, t_1)$, $move(r_1, b, c, t_2)$, $noOp(r_2, c, t_2)\}$. Arguments attacking A are in Table 1.[3] Arguments attacking $B_1 \ldots G_2$ (thus defending A) are shown in Table 2 (B_1'

[3] Here, m, n, nA and o are short-hands for $move, noOp, notAt$ and $occupied$, respectively.

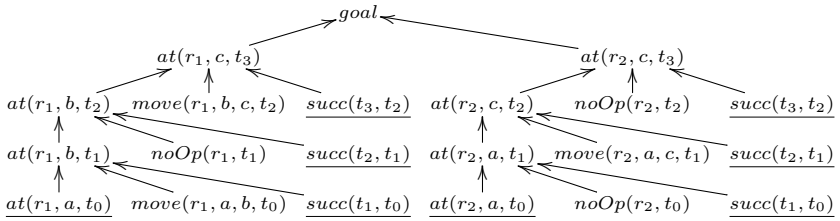

Fig. 2. An argument for $goal$ in Example 2. To save space, all leaf nodes τ, as the child of underlined nodes, are omitted.

attacks B_1, B'_2 attacks B_2, etc). We observe that all arguments in Table 2 are supported by assumptions in Δ except $B'_3 = \{nA(r_2, c, t_0)\} \vdash nA(r_2, c, t_0)$, $E'_2 = \{nA(r_1, a, t_1)\} \vdash nA(r_1, a, t_1)$ and $F'_3 = \{nA(r_2, b, t_2)\} \vdash nA(r_2, b, t_2)$. Among these, B'_3 is not attacked as there is no argument for $at(r_2, c, t_0)$. E'_2 is attacked by $H = \{n(r_1, a, t_0)\} \vdash at(r_1, a, t_1)$ and F'_3 is attacked by $I_1 = \{m(r_2, a, b, t_0), n(r_2, b, t_1)\} \vdash at(r_2, b, t_2)$, $I_2 = \{m(r_2, a, c, t_0), m(r_2, c, b, t_1)\} \vdash at(r_2, b, t_2)$, and argument $I_3 = \{n(r_2, a, t_0), m(r_2, a, b, t_1)\} \vdash at(r_2, b, t_2)$. However, H is attacked by B'_1, I_1 and I_2 are attacked by B'_2. I_3 is attacked by E'_1. Thus, A is defended by arguments in Table 2.

Table 2: Summary of arguments defending A in Example 2.

B'_1 $\{m(r_1, a, b, t_0)\} \vdash m(r_1, a, b, t_0)$	B'_2 $\{n(r_2, a, t_0)\} \vdash n(r_2, a, t_0)$
B'_3 $\{nA(r_2, c, t_0)\} \vdash nA(r_2, c, t_0)$	B'_4 $\{m(r_1, a, b, t_0)\} \vdash m(r_1, a, b, t_0)$
B'_5 $\{\} \vdash at(r_1, a, t_0)$	
C'_1 $\{n(r_2, a, t_0)\} \vdash n(r_2, a, t_0)$	C'_2 $\{n(r_2, a, t_0)\} \vdash n(r_2, a, t_0)$
D'_1 $\{n(r_1, b, t_1)\} \vdash n(r_1, b, t_1)$	D'_2 $\{n(r_1, b, t_1)\} \vdash n(r_1, b, t_1)$
E'_1 $\{m(r_2, a, c, t_1)\} \vdash m(r_2, a, c, t_1)$	E'_2 $\{nA(r_1, a, t_1)\} \vdash nA(r_1, a, t_1)$
E'_3 $\{n(r_1, b, t_1)\} \vdash n(r_1, b, t_1)$	E'_4 $\{m(r_2, a, c, t_1)\} \vdash m(r_2, a, c, t_1)$
E'_5 $\{n(r_2, a, t_0)\} \vdash at(r_2, a, t_1)$	E'_6 $\{m(r_1, a, b, t_0)\} \vdash m(r_1, a, b, t_0)$
E'_7 $\{m(r_1, a, b, t_0)\} \vdash m(r_1, a, b, t_0)$	
F'_1 $\{m(r_1, b, c, t_2)\} \vdash m(r_1, b, c, t_2)$	F'_2 $\{n(r_2, c, t_2)\} \vdash n(r_2, c, t_2)$
F'_3 $\{nA(r_2, b, t_2)\} \vdash nA(r_2, b, t_2)$	F'_4 $\{m(r_1, b, c, t_2)\} \vdash m(r_1, b, c, t_2)$
F'_5 $\{m(r_1, a, b, t_0), n(r_1, b, t_1)\} \vdash at(r_1, b, t_2)$	
G'_1 $\{n(r_2, c, t_2)\} \vdash n(r_2, c, t_2)$	G'_2 $\{n(r_2, c, t_2)\} \vdash n(r_2, c, t_2)$

Theorem 1. *Given a planning problem Π, let F_I be the instantiated framework corresponding to Π, then there is a valid plan for Π iff there exists an admissible argument $\Delta \vdash goal$ in F_I.*

Proof. (Sketch.) We first show that if a plan exists then an admissible argument $A = \Delta \vdash goal$. By Rule 1 in Definition 6, we know that to "prove" $goal$, we need to "prove" $at(X, L, t_n)$ for all blocks X, each at some location L. It is easy to see that using a combination of Rules 2 & 3 in Definition 5, all blocks can be placed to their specified locations (assuming t_n is large enough) so A can be constructed. To see that A is admissible, we make the following observations.

– Arguments targeting at $\alpha = noOp(\_,\_,\_)$ are of the form $\{move(\_,\_,\_,\_)\} \vdash move(\_,\_,\_,\_)$. These arguments can be counterattacked by $N = \{\alpha\} \vdash \alpha$. N does not attack A.
– Arguments targeting at $\alpha' = move(X, L, L', T)$ are of the following forms:
(1) $\{noOp(X, L, T)\} \vdash noOp(X, L, T)$ (not to move X away from L at time T) and $\{move(X, L, L'', T)\} \vdash move(X, L, L'', T)$ (move X to a different location). These arguments can be counterattacked by $M = \{\alpha'\} \vdash \alpha'$. M does not attack A.
(2) $\{move(X', L''', L', T)\} \vdash move(X, L''', L', T)$. These can be counterattacked by either $\{move(X', L''', L^*, T)\} \vdash move(X, L''', L^*, T)$ or $\{noOp(X', L''', T)\} \vdash noOp(X', L''', T)$ or $\{notAt(X', L''', T)\} \vdash notAt(X', L''', T)$. These arguments do not attack A.
(3) $\_ \vdash above(X', X, L, T)$. By Rule 1 in Definition 5, to have arguments of this form, we need to have some block X' at the same location as X but smaller. Under such cases, X should not be moved thus no such $move(X, L, L', T)$ would be used to support A.
(4) $\_ \vdash occupied(X', X, T)$. By Rule 4 in Definition 5, to have arguments of this form, we have some block smaller than X at the destination of the move. In such cases, X should not be moved to that destination so no such $move(X, L, L', T)$ would be used to support A.

Since a plan exists for this problem, a sequence of $move$s and $noOp$s, which would not trigger indefensible attacks from $\Delta \vdash above(X', X, L, T)$ or $\Delta \vdash occupied(X', X, T)$ must exist. The other direction of this theorem is trivial as once an admissible A is found, assumptions from A consist a plan.

The following corollary follows trivially from Theorem 1.

Corollary 1. *Given a planning problem Π, let F_I be the instantiated framework corresponding to Π, if $\Delta \vdash goal$ is admissible in F_I, then Δ is a valid plan for Π.*[4]

Theorem 1 and Corollary 1 establish the connection between planning in blocks world and ABA frameworks. The admissibility of the argument A for $goal$ can be viewed as a means to justify the "validity" of the plan as every assumption supporting the argument A is "defended". This can be read as every action in the plan is valid. Similarly, non-admissible arguments of the form $\Delta \vdash goal$ correspond to invalid plans, illustrated with the next example.

Example 3. Given the blocks world shown in Figure 1, the plan

$$P = \{move(r_1, a, c, t_0), noOp(r_2, a, t_0), move(r_2, a, c, t_1), noOp(r_2, c, t_1)\}$$

is invalid as c is occupied by r_1 at t_1 and $r_1 < r_2$. So $move(r_2, a, c, t_1)$ is invalid and it is in an explanation for P. Let F_I be the instantiated framework.

$$A' = \{move(r_1, a, c, t_0), noOp(r_2, a, t_0), move(r_2, a, c, t_1), noOp(r_2, c, t_1)\} \vdash goal$$

[4] We abuse the notation Δ. Here and hereinafter, Δ is used to represent both a set of assumptions in the instantiated framework F_I and a plan containing a set of actions with syntactically identical names in the corresponding planning problem Π.

is not admissible in F_I. Arguments attacking A' are shown in Table 3. Using reasoning similar to Example 2, we see that A is able to defend all of its attackers except E_6 as E_6 is supported by a single assumption $m(r_1, a, c, t_0)$, which also supports A'. Thus, any argument B_i attacks E_6 must also attack A'. Any set of argument containing A', B_i cannot be conflict-free, therefore A' is not admissible.

Table 3: Arguments attacking A' in Example 3.

Arguments targeting at $move(r_1, a, c, t_0)$:	
$B_1 = \{m(r_1, a, b, t_0)\} \vdash m(r_1, a, b, t_0)$	$B_2 = \{m(r_2, a, c, t_0)\} \vdash m(r_2, a, c, t_0)$
$B_3 = \{m(r_2, b, c, t_0)\} \vdash m(r_1, b, c, t_0)$	$B_4 = \{n(r_1, a, t_0)\} \vdash n(r_1, a, t_0)$
$B_5 = \{nA(r_1, a, t_0)\} \vdash nA(r_1, a, t_0)$	
Arguments targeting at $noOp(r_2, a, t_0)$:	
$C_1 = \{m(r_2, a, b, t_0)\} \vdash m(r_2, a, b, t_0)$	$C_2 = \{m(r_2, a, c, t_0)\} \vdash m(r_2, a, c, t_0)$
Arguments targeting at $noOp(r_1, c, t_1)$:	
$D_1 = \{m(r_1, c, a, t_1)\} \vdash m(r_1, c, a, t_1)$	$D_2 = \{m(r_1, c, b, t_1)\} \vdash m(r_1, c, b, t_1)$
Arguments targeting at $move(r_2, a, c, t_1)$:	
$E_1 = \{m(r_2, a, b, t_1)\} \vdash m(r_2, a, b, t_0)$	$E_2 = \{m(r_1, a, c, t_1)\} \vdash m(r_1, a, c, t_1)$
$E_3 = \{m(r_1, b, c, t_1)\} \vdash m(r_1, b, c, t_1)$	$E_4 = \{n(r_2, a, t_1)\} \vdash n(r_2, a, t_1)$
$E_5 = \{nA(r_2, a, t_1)\} \vdash nA(r_2, a, t_1)$	$E_6 = \{m(r_1, a, c, t_0)\} \vdash o(r_2, c, t_1)$
$E_7 = \{n(r_1, a, t_0), n(r_2, a, t_0)\} \vdash above(r_1, r_2, a, t_1)$	

5 Conclusion

In this paper, we studied how to use ABA to model planning problems in line with the *defeasible planning* proposal suggested by Pollock. Using blocks world as a case study, we demonstrated the feasibility of using ABA to plan. The two key ideas are (1) with actions modelled with assumptions, plan construction can be equated to the construction of ABA arguments and (2) by modelling action constraints as arguments attacking the plan, identifying valid plans can be equated to computing admissible ABA arguments. In future, we will generalise this work to create argumentation-based planning models, study its connection with situation calculus, SAT planning, or BDD-based symbolic planning and apply our work in some real-world practical planning applications.

References

1. L. Amgoud, C. Devred, and M. Lagasquie-Schiex. A constrained argumentation system for practical reasoning. In *Prof. of AAMAS*, pages 429–436, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
2. L. Amgoud, C. Devred, and M. Lagasquie-Schiex. Generating possible intentions with constrained argumentation systems. *IJAR*, 52(9):1363–1391, 2011.
3. K. Čyras, X. Fan, C. Schulz, and F. Toni. Assumption-based argumentation: Disputes, explanations, preferences. *IfCoLog JLTA*, 4(8), 2017.
4. 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.
5. D. Nau, M. Ghallab, and P. Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
6. J. Pollock. Defeasible planning. In *Proc. of AAAI Workshop, Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments*, 1998.