Value-Based Practical Reasoning: Modal Logic + Argumentation

Jieting Luo [a,1] Beishui Liao [b] and Dov Gabbay [c]

[a] *University of Bern, Swtzerland*
[b] *Zhejiang University, China*
[c] *King's College London, UK*

Abstract. Autonomous agents are supposed to be able to finish tasks or achieve goals that are assigned by their users through performing a sequence of actions. Since there might exist multiple plans that an agent can follow and each plan might promote or demote different values along each action, the agent should be able to resolve the conflicts between them and evaluate which plan he should follow. In this paper, we develop a logic-based framework that combines modal logic and argumentation for value-based practical reasoning with plans. Modal logic is used as a technique to represent and verify whether a plan with its local properties of value promotion or demotion can be followed to achieve an agent's goal. We then propose an argumentation-based approach that allows an agent to reason about his plans in the form of supporting or objecting to a plan using the verification results.

Keywords. Value, Practical Reasoning, Modal Logic, Argumentation

1. Introduction

Autonomous agents are supposed to be able to perform value-based ethical reasoning based on their value systems in order to distinguish moral from immoral behavior. Existing work on value-based practical reasoning such as [1][2] [3] demonstrates how an agent can reason about what he should do among alternative action options that are associated with value promotion or demotion. However, agents are supposed to be able to finish tasks or achieve goals that are assigned by their users through performing a sequence of actions. Since there might exist multiple plans that an agent can follow and each plan might promote or demote different values along each action, the agent should be able to resolve the conflicts between them and evaluate which plan he should follow. If the decision-making problem concerns choosing a plan instead of an action, then we first need to know how an agent can see whether he can follow a plan to achieve his goal. Verification approaches that are developed based on modal logic only allow us to verify whether a goal can be achieved under specific conditions such as norm compliance assumptions [4][5][6], namely telling us whether a plan works or not, but cannot tell us what we should do. For sure, we can collect the verification results regarding whether a plan promotes or demotes a specific set of values and then compare different plans using

---

[1]Corresponding Author.

lifting approaches as what has been done in [7]. However, the order lifting problem is a major challenge in many areas of AI and no approach is ultimately "correct". Moreover, the agent in our setting needs to lift the preference over values to the preference over plans with respect to value promotion and demotion, which even complicates the problem. Therefore, we need a more natural and intuitive approach. It has been shown that argumentation provides a useful mechanism to model and resolve conflicts [8], and particularly can be used for the decision-making of artificial intelligence and provides explanation for that [9][10]. In this paper, we develop a logic-based framework that combines modal logic and argumentation for value-based practical reasoning with plans. Modal logic is used as a technique to represent and verify whether a plan with its local properties of value promotion or demotion can be followed to achieve an agent's goal. Using the verification results to construct arguments, we then propose an argumentation-based approach that allows an agent to reason about his plans in the form of support and objection without using lifting approaches. We prove several formal properties to characterize our approach, indicating it is consistent with our rationality of decision-making.

2. Logical Framework

The semantic structure of this paper is a transition system that represents the computational behavior of a system caused by an agent's actions in the agent's subjective view. It is basically a directed graph where a set of vertexes S corresponds to possible states of the system, and the relation $\rightarrow \subseteq S \times Act \times S$ represents the possible transitions of the system. When a certain action $\alpha \in Act$ is performed, the system might progress from a state s to a different state s' in which different propositions hold. Formally,

Definition 1 (Transition Systems). *Let* $\Phi = \{p, q, \ldots\}$ *be a finite set of atomic propositional variables, a transition system is a tuple* $T = (S, Act, \rightarrow, V)$ *over* Φ*, where*

- *S is a finite, non-empty set of states;*
- *Act is a finite, non-empty set of actions;*
- *$\rightarrow \subseteq S \times Act \times S$ is a transition relation between states with actions, which we refer to as the transition relation labeled with an action; we require that for all $s \in S$ there exists an action $a \in Act$ and a state $s' \in S$ such that $(s, a, s') \in \rightarrow$; we restrict actions to be deterministic, that is, if $(s, a, s') \in \rightarrow$ and $(s, a, s'') \in \rightarrow$, then $s' = s''$; since the relation is partially functional, we write $s[\alpha]$ to denote the state s' for which it holds that $(s, \alpha, s') \in \rightarrow$; we also use $s[\alpha_1, \ldots, \alpha_n]$ to denote the resulting state for which a sequence of actions $\alpha_1, \ldots, \alpha_n$ succinctly execute from state s;*
- *V is a propositional valuation $V : S \rightarrow 2^{\Phi}$ that assigns each state with a subset of propositions which are true at state s; thus for each $s \in S$ we have $V(s) \subseteq \Phi$.*

Note that the model is deterministic: the same action performed in the same state will always result in the same resulting state. A pointed transition system is a pair (T, s) such that T is a transition system, and $s \in S$ is a state from T. Adopted from [11][12], the language $\mathcal{L}$ is propositional logic extended with action modality. Formally, its grammar is defined below:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid Do(\alpha)\varphi \quad (p \in \Phi, \alpha \in Act)$$
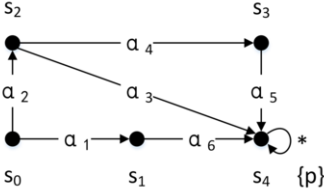
Figure 1. Transition system T.
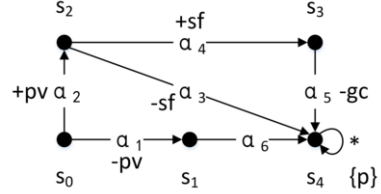
Figure 2. A Value-based transition system VT.

Given a pointed transition system (T,s), we define the semantics with respect to the satisfaction relation $\vDash$ inductively as follows:

- $T,s \vDash p$ iff $p \in V(s)$;
- $T,s \vDash \neg\varphi$ iff $T,s \nvDash \varphi$;
- $T,s \vDash \varphi \vee \psi$ iff $T,s \vDash \varphi$ or $T,s \vDash \psi$;
- $T,s \vDash Do(\alpha)\varphi$ iff $s[\alpha] \vDash \varphi$.

The remaining classical logic connectives are assumed to be defined as abbreviations in terms of $\neg$ and $\vee$ in the conventional manner. Given a pointed transition system (T,s), we say that a sequence of actions $\alpha_1 \ldots \alpha_n$ brings about a φ-state if and only if $T,s \vDash Do(\alpha_1)\ldots Do(\alpha_n)\varphi$. As standard, we write $T \vDash \varphi$ if $T,s \vDash \varphi$ for all $s \in S$, and $\vDash \varphi$ if $T \vDash \varphi$ for all T.

A transition system represents how a system progresses by an agent's actions. Besides, an agent in the system is assumed to have his own goal, which is a formula expressed in propositional logic $\mathcal{L}_{prop}$. It is indeed possible for an agent to have multiple goals and his preference over different goals. For example, a goal hierarchy is defined in [4] to represent increasingly desired properties that the agent wishes to hold. However, we find that the setting about whether the agent has a goal or multiple goals is in fact not essential for our analysis, so we simply assume that the agent only has a goal for simplifying our presentation.

Example 1. *Consider the transition system T in Figure 1, which represents how an agent can get to a pharmacy to buy medicine for his user. State s_0 is the initial state, representing staying at home, and proposition p, representing arriving at a pharmacy, holds in state s_4. The agent can perform actions α_1 to α_6 in order to get to state s_4. From this transition system, the following formulas hold:*

$$T,s_0 \vDash Do(\alpha_1)Do(\alpha_6)p,$$

$$T,s_0 \vDash Do(\alpha_2)Do(\alpha_3)p,$$

$$T,s_0 \vDash Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p,$$

which means that the agent can first perform action α_1 and then action α_6, or action α_2 followed by action α_4, or action α_2 followed by actions α_4 and α_5, to get to the pharmacy.

It is important for an agent not only to achieve his goal, but also to think about how to achieve his goal. As we can see from the running example, there are multiple ways for

the agent to get to the pharmacy, and the agent needs to evaluate which one is the best to choose. In this paper, agents are able to perform value-based practical reasoning in terms of planning their actions to achieve their goals. We first assume that an agent has a set of values. A value can be seen as an abstract standard according to which agents have their preferences over options. For instance, if we have a value denoting *equality*, we prefer the options where equal sharing or equal rewarding hold. Unlike [7] where a value is interpreted as a state formula, we simply assume a value as a primitive structure without considering how it is defined. We assume that agents can always compare any two values, so we define an agent's value system as a total pre-order (instead of a strict total order) over a set of values, representing the degree of importance of something.

Definition 2 (Value System). *A value system $V = (\mathrm{Val}, \lesssim)$ is a tuple consisting of a finite set of values $\mathrm{Val} = \{v, ..., v'\}$ together with a total pre-ordering $\lesssim$ over Val. When $v \lesssim v'$, we say that value v' is at least as important as value v. As is standard, we define $v \sim v'$ to mean $v \lesssim v'$ and $v' \lesssim v$, and $v < v'$ to mean $v \lesssim v'$ and $v \not\gtrsim v'$.*

We label some of the transitions with the values promoted and demoted by moving from a starting state to a ending state. Notice that not every transition can be labeled, as some transitions may not be relevant to any value in an agent's value system. Formally, function $\delta : \{+, -\} \times \mathrm{Val} \to 2^{\to}$ is a valuation function which defines the status (promoted (+) or demoted (-)) of a value $v \in \mathrm{Val}$ ascribed to a set of transitions. We then define a value-based transition system VT as a transition system together with a value system V and a function δ.

Definition 3 (Value-based Transition Systems). *A value-based transition system is defined by a triple $VT = (T, V, \delta)$, where T is a transition system, V is a value system and δ is a valuation function that assigns value promotion or demotion to a set of transitions.*

Given a sequence of actions with respect to a value-based transition system, we then express whether the performance of the sequence in a state promotes or demotes a specific value, which can be done by extending our language. Given a pointed value-based transition system (VT, s) and a value $v \in \mathrm{Val}$, the satisfaction relation $VT, s \vDash \psi$ is extended with the following new semantics:

- $VT, s \vDash_{+v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ iff $s[\alpha_1, \ldots, \alpha_n] \vDash \varphi$ and there exists $1 \le m \le n$ such that $(s[\alpha_1, \ldots, \alpha_{m-1}], \alpha_m, s[\alpha_1, \ldots, \alpha_m]) \in \delta(+, v)$;
- $VT, s \vDash_{-v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ iff $s[\alpha_1, \ldots, \alpha_n] \vDash \varphi$ and there exists $1 \le m \le n$ such that $(s[\alpha_1, \ldots, \alpha_{m-1}], \alpha_m, s[\alpha_1, \ldots, \alpha_m]) \in \delta(-, v)$.

The formula $VT, s \vDash_{+v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ (resp. $VT, s \vDash_{-v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$) should be intuitively read as φ is achieved after the performance of a sequence of actions $\alpha_1, \ldots, \alpha_n$ in state s and there exists an action that promotes (resp. demotes) value v in the sequence. Notice that the formula only expresses the local property of a sequence of actions in terms of value promotion or demotion by an action within the sequence. Thus, it is possible that an action within the sequence promotes value v but it gets demoted by another action within the sequence, meaning that both $VT, s \vDash_{+v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ and $VT, s \vDash_{-v} Do(\alpha_1), \ldots, Do(\alpha_n)\varphi$ hold at the same time. Through checking the above formulas, the agent is then aware of whether he can perform the sequence of actions to achieve his goal and which value gets promoted or demoted along the sequence. We con-

tinue our running example to illustrate how to use our logical language to express and verify properties of sequences of actions.

Example 2. *Suppose the ethical agent has privacy (pv), safety (sf) and good conditions (gc) as his values and a value system as $pv \prec gc \prec sf$. As in Figure 2, some of the transitions have been labeled with value promotion or demotion with respect to the agent's values. Taking action α_1 in state s_0 is interpreted as going through the neighbor's garden for taking shortcut, which demotes the value of privacy of the neighbor, conversely taking action α_2 in state s_0 is interpreted as stepping on a normal way, which promotes the value of privacy of the neighbor. Taking action α_3 means crossing the road without using the crosswalk, which demotes the value of safety of the agent, and conversely taking action α_4 in state s_2 promotes the value of safety of the agent. Finally, performing action α_5 in state s_3 means stepping into water. As the agent is a robot, which should avoid getting wet, this choice will demote the value of maintaining good conditions of the agent. The agent can verify whether he can achieve his goal while promoting or demoting a specific value by performing a sequence of actions. The verification results are listed below:*

$$VT, s_0 \vDash_{-pv} Do(\alpha_1)Do(\alpha_6)p$$

$$VT, s_0 \vDash_{+pv} Do(\alpha_2)Do(\alpha_3)p$$

$$VT, s_0 \vDash_{-sf} Do(\alpha_2)Do(\alpha_3)p$$

$$VT, s_0 \vDash_{+pv} Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p$$

$$VT, s_0 \vDash_{+sf} Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p$$

$$VT, s_0 \vDash_{-gc} Do(\alpha_2)Do(\alpha_4)Do(\alpha_5)p$$

3. Planning: an Argumentation-based Approach

Given a transition system and an agent's goal, model checking and verification techniques allow us to verify whether an agent can achieve his goal while promoting or demoting a specific value by performing a sequence of actions. Since following different plans might promote or demote different sets of values, next question is how the agent decides what to do given the verification results. In this paper, we propose to use argumentation as a technique for an agent's decision-making. Formal argumentation is a nonmonotonic formalism for representing and reasoning about conflicts based on the construction and the evaluation of interacting arguments [8]. In particular, it has been used in practical reasoning, which is concerned by reasoning about what agents should do, given different alternatives and outcomes they bring about [2][10]. We first define the notion of plans. A plan is defined as a finite sequence of actions that are enabled by our underlying transition system. Formally,

Definition 4 (Plans)**.** *Given a pointed value-based transition system (VT, s) and a formula $p \in \mathcal{L}_{prop}$ as an agent's goal, a plan is defined as a finite sequence of actions over Act, denoted as $\lambda = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, such that $VT, s \vDash Do(\alpha_1)Do(\alpha_2)\ldots Do(\alpha_n)p$.*

The definition is equivalent to saying that a plan is a sequence of actions, each of which can be performed succinctly with respect to the pointed value-based transition

system, to achieve the agent's goal. The agent has to reason about the available plans with respect to their goal achievement and value promotion or demotion. In order to do that, it is intuitive to define an argument as a plan together with its local property of value promotion or demotion. Based on the verification results, we define two types of arguments.

Definition 5 (Ordinary Arguments and Blocking Arguments)**.** *Given a pointed value-based transition system* (VT, s)*, a formula* $p \in \mathcal{L}_{prop}$ *as an agent's goal, a plan* $\lambda = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ *and* $v \in$ Val*, an ordinary argument is a pair* $\langle +v, \lambda \rangle$ *such that* $VT, s \vDash_{+v} Do(\alpha_1)Do(\alpha_2)\ldots Do(\alpha_n)p$*; a blocking argument is a pair* $\langle -v, \neg\lambda \rangle$ *such that* $VT, s \vDash_{-v} Do(\alpha_1)Do(\alpha_2)\ldots Do(\alpha_n)p$*; we use* $\mathcal{A}_o$ *(resp.* $\mathcal{A}_b$*) to denote the set of ordinary arguments (resp. blocking arguments), and* $\mathcal{A} = \mathcal{A}_o \cup \mathcal{A}_b$ *to denote the set of two types of arguments.*

Both an ordinary argument and a blocking argument correspond to a verification result. An ordinary argument $\langle +v, \lambda \rangle$ is interpreted as "the agent should follow plan λ to achieve his goal because it promotes a value v", which supports the performance of plan λ, and a blocking argument $\langle -v, \neg\lambda \rangle$ is interpreted as "the agent should not follow plan λ to achieve his goal because it demotes a value v", which objects to the performance of plan λ. Conventionally, we might represent an argument using an alphabet $(a, b, \ldots)$ if we do not care about the internal structure of the argument.

Example 3. *From the verification results listed in Example 2, the agent can construct the following arguments:* $\langle -pv, \neg(\alpha_1, \alpha_6) \rangle$*,* $\langle +pv, (\alpha_2, \alpha_3) \rangle$*,* $\langle -sf, \neg(\alpha_2, \alpha_3) \rangle$*,* $\langle +pv, (\alpha_2, \alpha_4, \alpha_5) \rangle$*,* $\langle +sf, (\alpha_2, \alpha_4, \alpha_5) \rangle$ *and* $\langle -gc, \neg(\alpha_2, \alpha_4, \alpha_5) \rangle$*.*

When we get to choose a plan to follow, there are conflicts between the alternatives as they cannot be followed all at the same time. The conflicts are interpreted as attacks between two ordinary arguments supporting different plans and one ordinary argument and one blocking argument supporting and objecting to the same plan respectively in this paper.

Definition 6 (Attacks)**.** *Given a set of ordinary arguments* $\mathcal{A}_o$ *and a set of blocking arguments* $\mathcal{A}_b$*,*

- *for any two ordinary arguments* $\langle +v_a, \lambda_a \rangle, \langle +v_b, \lambda_b \rangle \in \mathcal{A}_o$*,* $\langle +v_a, \lambda_a \rangle$ *attacks* $\langle +v_b, \lambda_b \rangle$ *iff* $\lambda_a \neq \lambda_b$*;*
- *for any ordinary argument* $\langle +v_a, \lambda_a \rangle \in \mathcal{A}_o$ *and any blocking argument* $\langle -v_b, \neg\lambda_b \rangle \in \mathcal{A}_b$*,*
  * $\langle +v_a, \lambda_a \rangle$ *attacks* $\langle -v_b, \neg\lambda_b \rangle$ *iff* $\lambda_a = \lambda_b$*;*
  * $\langle -v_b, \neg\lambda_b \rangle$ *attacks* $\langle +v_a, \lambda_a \rangle$ *iff* $\lambda_a = \lambda_b$*.*

The set of attacks over $\mathcal{A}$ *are denoted as* $\mathcal{R}$*.*

It is obvious that our attack relation is mutual. It should be noticed that there is no attack between two blocking arguments, as a blocking argument only functions as blocking the conclusion of an ordinary argument but does not make a conclusion by itself.

The attack relation represents conflicts between plans. However, the notion of attack may not be sufficient for modeling conflicts between arguments, as an agent has his

preference over the values that are promoted or demoted by different plans. In structured argumentation frameworks such as *ASPIC$^+$* [13], an argument *a* can be used as a counter-argument to another argument *b*, if *a* successfully attacks, i.e. defeats, *b*. Whether an attack from *a* to *b* (on its sub-argument *b'*) succeeds as a defeat, may depend on the relative strengths of *a* and *b*, i.e. whether *a* is strictly stronger than, or strictly preferred over *b'*. For this paper, recall that an agent has a value system, which was defined as a total pre-order over a set of values. We can then determine the preference over two arguments with respect to value promotion and demotion based on the value system. The notion of defeats combines the notions of attack and preference.

Definition 7 (Defeats). *Given a set of ordinary arguments $\mathcal{A}_o$ and a set of blocking arguments $\mathcal{A}_b$, a set of attacks $\mathcal{R}$ over $\mathcal{A}$ and a value system V,*

- *for any two ordinary arguments $\langle +v_a, \lambda_a \rangle, \langle +v_b, \lambda_b \rangle \in \mathcal{A}_o$, $\langle +v_a, \lambda_a \rangle$ defeats $\langle +v_b, \lambda_b \rangle$ iff $\langle +v_a, \lambda_a \rangle$ attacks $\langle +v_b, \lambda_b \rangle$ and $v_a \nless v_b$;*
- *for any ordinary argument $\langle +v_a, \lambda_a \rangle \in \mathcal{A}_o$ and any blocking argument $\langle -v_b, \neg\lambda_b \rangle \in \mathcal{A}_b$,*

  * *$\langle +v_a, \lambda_a \rangle$ defeats $\langle -v_b, \neg\lambda_b \rangle$ iff $\langle +v_a, \lambda_a \rangle$ attacks $\langle -v_b, \neg\lambda_b \rangle$ and $v_a \nless v_b$;*
  * *$\langle -v_b, \neg\lambda_b \rangle$ defeats $\langle +v_a, \lambda_a \rangle$ iff $\langle -v_b, \neg\lambda_b \rangle$ attacks $\langle +v_a, \lambda_a \rangle$ and $v_b \nless v_a$.*

The set of defeats over $\mathcal{A}$ based on an attack relation and a value system are denoted as $\mathcal{D}(\mathcal{R}, V)$. We write $\mathcal{D}$ for short if it is clear from the context.

In words, given mutual attacks between two arguments, the attack from the argument with less preferred value to the attack from the argument with a more preferred value does not succeed as a defeat. One might ask whether it is more convenient to combine the notions of attack relation and defeat relation. We argue that two notions represent the relation between two arguments from different perspectives, one for the conflicts between plans and the other for the preferences over values. Because of that, defining these two notions separately can make our framework more clear, even though technically it is possible to combine them. It is obvious to see that our defeat relation can form a *two-length cycle* in which two arguments have equivalent or the same values.

Proposition 1. *Given two ordinary arguments $\langle +v_a, \lambda_a \rangle, \langle +v_b, \lambda_b \rangle \in \mathcal{A}_o$, $\langle +v_a, \lambda_a \rangle$ and $\langle +v_b, \lambda_b \rangle$ form a two-length cycle iff $\lambda_a \neq \lambda_b$ and ($v_a = v_b$ or $v_a \sim v_b$). Given an ordinary argument $\langle +v_a, \lambda_a \rangle \in \mathcal{A}_o$ and a blocking argument $\langle -v_b, \neg\lambda_b \rangle \in \mathcal{A}_b$, $\langle +v_a, \lambda_a \rangle$ and $\langle -v_b, \neg\lambda_b \rangle$ form a two-length cycle iff $\lambda_a = \lambda_b$ and ($v_a = v_b$ or $v_a \sim v_b$).*

Proof. Proof follows from Definition 7 directly. □

However, we have the result that our defeat relation obeys the property of irreflexivity and never forms any odd cycle.

Proposition 2. *Given a set of arguments $\mathcal{A}$, a defeat relation $\mathcal{D}$ on $\mathcal{A}$ never forms any odd cycle.*

Proof. According to Definition 7, in order for an argument *a* to defeat another argument *b*, the value v_a that belongs to *a* must be not less preferred than v_b that belongs to *b*. Since an agent's value system is a total pre-order over a set of values, arguments can only form a cycle in which any two arguments are mutually defeated with the values involved are equivalent or the same. So $\mathcal{D}$ on $\mathcal{A}$ never forms any odd cycle. □
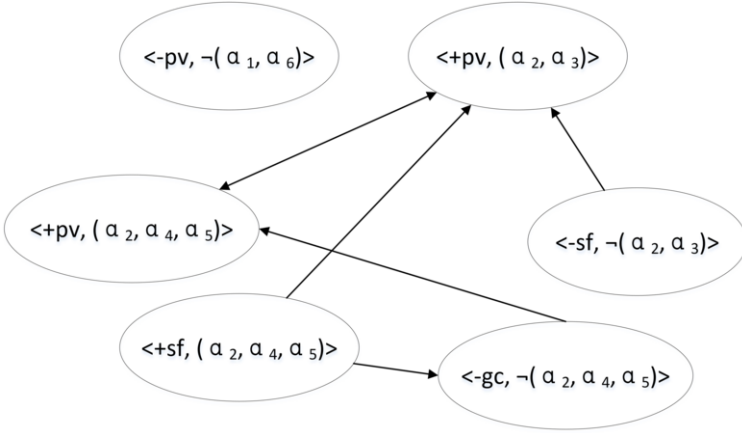
Figure 3. An argumentation framework.

Proposition 3. *Given a set of arguments $\mathcal{A}$, a defeat relation $\mathcal{D}$ on $\mathcal{A}$ is irreflexive.*

Proof. It is a special case of Proposition 2 for the number of arguments in the odd cycle being one. □

We are now ready to construct a Dung-style abstract argumentation framework with ordinary arguments, blocking arguments and the defeat relation on them.

Definition 8 (Plan-based Argumentation Frameworks)**.** *Given a pointed value-based transition system (VT, s) and a formula $p \in \mathcal{L}_{prop}$ as an agent's goal, a plan-based argumentation framework over (VT, s) and p is a pair $PAF = (\mathcal{A}, \mathcal{D})$, where $\mathcal{A}$ is a set of arguments and $\mathcal{D}$ is a defeat relation on $\mathcal{A}$.*

Example 4. *The running example has three ordinary arguments and three blocking arguments, and any two ordinary arguments with different plans are mutually attacked, and any ordinary argument and blocking argument with the same plan are mutually attacked. Suppose the agent has a value system as $pv \prec gc \prec sf$, which means that safety is more important than keeping good condition, and keeping good condition is more important than privacy. We then can see some of the attacks do not succeed as defeats. For example, argument $\langle +pv, (\alpha_2, \alpha_4, \alpha_5) \rangle$ and argument $\langle -gc, \neg(\alpha_2, \alpha_4, \alpha_5) \rangle$ are mutually attacked, but since pv is less preferred than gc, only the attack from argument $\langle -gc, \neg(\alpha_2, \alpha_4, \alpha_5) \rangle$ to argument $\langle +pv, (\alpha_2, \alpha_4, \alpha_5) \rangle$ becomes a defeat. For arguments $\langle +pv, (\alpha_2, \alpha_4, \alpha_5) \rangle$ and $\langle +pv, (\alpha_2, \alpha_3) \rangle$, since $pv = pv$, the mutual attacks between them succeed as mutual defeats. For the space limitation, we do not analyze all the defeats, which is depicted in Figure 3. Interestingly, argument $\langle -pv, \neg(\alpha_1, \alpha_6) \rangle$ does not receive any defeats or defeat any arguments not because pv is the most preferred value, but because there is no ordinary argument with plan (α_1, α_6).*

Given a plan-based argumentation framework *PAF*, status of arguments is evaluated, producing sets of arguments that are acceptable together, which are based on the notions of conflict-freeness, acceptability and admissibility. The well-known argumentation semantics are listed as follows, each of which provides a pre-defined criterion for determining the acceptability of arguments in a *PAF* [8].

Definition 9 (Conflict-freeness, Acceptability, Admissibility and Extensions). *Given PAF* $= (\mathcal{A}, \mathcal{D})$ *and* $E \subseteq \mathcal{A}$,

- *A set E of arguments is conflict-free iff there does not exist* $a, b \in E$ *such that* $(a, b) \in \mathcal{D}$.
- *An argument* $a \in \mathcal{A}$ *is acceptable w.r.t. a set E (a is defended by E), iff* $\forall (b, a) \in \mathcal{D}$, $\exists c \in E$ *such that* $(c, b) \in \mathcal{D}$.
- *A conflict-free set of arguments E is admissible iff each argument in E is acceptable w.r.t. E.*
- *E is a complete extension of PAF iff E is admissible and each argument in* $\mathcal{A}$ *that is acceptable w.r.t. E is in E.*
- *E is the grounded extension of PAF iff E is the minimal (w.r.t. set inclusion) complete extension.*
- *E is the preferred extension of PAF iff E is a maximal (w.r.t. set inclusion) complete extension.*
- *E is a stable extension of PAF iff E is conflict-free and* $\forall b \in \mathcal{A} \backslash E$, $\exists a \in E$ *such that* $(a, b) \in \mathcal{D}$.

We use $sem \in \{cmp, prf, grd, stb\}$ to denote the complete, preferred, grounded and stable semantics, respectively, and $\mathcal{E}_{sem}(PAF)$ to denote the set of extensions of *PAF* under a semantics in *sem*. The following propositions characterize our argumentation framework in terms of Dung's semantics.

Proposition 4. *Given PAF* $= (\mathcal{A}, \mathcal{D})$, $\mathcal{E}_{prf}(PAF) = \mathcal{E}_{stb}(PAF)$.

Proof. Since our defeat relation $\mathcal{D}$ never forms an odd cycle by Proposition 2, which means that *PAF* is limited controversial, each preferred extension of *PAF* is stable. Detailed proof can be found in [8]. ☐

Proposition 5. *Given PAF* $= (\mathcal{A}, \mathcal{D})$, *if there exists an ordinary argument* $\langle +v_a, \lambda_a \rangle$ *such that for all* $\langle +v_b, \lambda_b \rangle \in \mathcal{A}_o$ *and* $\langle -v_b, \neg\lambda_b \rangle \in \mathcal{A}_b$ *it is the case that* $v_b \precsim v_a$ *and* $\langle +v_a, \lambda_a \rangle$ *is not in any cycle, then* $\mathcal{E}_{prf}(PAF) = \mathcal{E}_{grd}(PAF)$.

Proof. Because $v_b \precsim v_a$ and $\langle +v_a, \lambda_a \rangle$ is not in any cycle, argument $\langle +v_a, \lambda_a \rangle$ does not receive any defeats. So the grounded extension is not an empty set. Suppose $\mathcal{E}_{prf}(PAF) \neq \mathcal{E}_{grd}(PAF)$, which means that there are more than one preferred extensions. Since $\langle +v_a, \lambda_a \rangle$ is contained in the grounded extension, it should also be contained in each preferred extension. However, each preferred extension indicates a distinct plan, which will be later proved by Proposition 6 and its implication. Contradiction! ☐

The justification of optimal plans is then defined under various semantics in Definition 9. Similarly to [14], we write concl($\langle +v, \lambda \rangle$) for the conclusion λ of an ordinary argument, and Oplans(*PAF, sem*) for the set of conclusions of ordinary arguments from the extensions under a specific semantics.

Definition 10 (Optimal Plans). *Given PAF* $= (\mathcal{A}, \mathcal{D})$, *a set of optimal plans, written as* Oplans(*PAF, sem*), *are the conclusions of the ordinary arguments within extensions.*

$$\text{Oplans}(PAF, sem) = \{\text{concl}(\langle +v, \lambda \rangle) \mid \langle +v, \lambda \rangle \in E \text{ and } E \in \mathcal{E}_{sem}(PAF)\}$$

We show that the results of our approach are consistent with the rationality of decision-making through the following propositions. Firstly, all the accepted arguments within an extension indicate the same plan.

Proposition 6. *Given a plan-based argumentation framework PAF = $(\mathcal{A}, \mathcal{D})$ and an extension E of PAF under a specific semantics as defined in Definition 9,*

1. *for any two ordinary arguments $\langle +v_a, \lambda_a \rangle, \langle +v_b, \lambda_b \rangle \in E$, it is the case that $\lambda_a = \lambda_b$;*
2. *for any ordinary argument $\langle +v_a, \lambda_a \rangle \in E$ and any blocking argument $\langle -v_b, \neg \lambda_b \rangle \in E$, $\lambda_a \neq \lambda_b$.*

Proof. For any extension E under a specific semantics, it is required that all the arguments in E should be conflict-free. 1. By Definition 7, we can derive two cases: either there is no attack between these two arguments, or one argument attacks the other but does not succeed as a defeat due to the preference between two values from the arguments. For the former case, two arguments contain the same plan. For the latter case, since any attack between two arguments is mutual, if an attack from argument $\langle +v_a, \lambda_a \rangle$ to argument $\langle +v_b, \lambda_b \rangle$ fails to be a defeat due to the preference between two values from the arguments, the attack from argument $\langle +v_b, \lambda_b \rangle$ to argument $\langle +v_a, \lambda_a \rangle$ will succeed to be a defeat. That means that the second case is impossible and only the first case holds. Hence, the two arguments have the same plan. 2. We can prove in a similar way that for any ordinary argument $\langle +v_a, \lambda_a \rangle \in E$ and any blocking argument $\langle -v_b, \neg \lambda_b \rangle \in E$, $\lambda_a \neq \lambda_b$, □

From that we can see, if there are multiple preferred extensions, then each of them indicates a distinct plan. Secondly, our argumentation-based approach always selects the plan through which the most preferred value gets promoted and does not select the plan through which the most preferred value gets demoted.

Proposition 7. *Given a plan-based argumentation framework PAF = $(\mathcal{A}, \mathcal{D})$, let $v_a \in$ Val be a value such that for all $\langle +v_b, \lambda_b \rangle \in \mathcal{A}_o$ and $\langle -v_b, \neg \lambda_b \rangle \in \mathcal{A}_b$ it is the case that $v_b \precsim v_a$, then an argument with value v_a is in a preferred extension. Typically, if it is not in a cycle, then it is in the grounded extension.*

Proof. Because $v_b \precsim v_a$, according to Definition 7, an argument with v_a only gets defeated by an argument with $v_a \sim v_b$ or $v_a = v_b$. In such a case, the defeats are mutual so $\langle +v_a, \lambda_a \rangle$ is self-defended. Thus, it is contained in a preferred extension. If it is not in a cycle, which means that it is not self-defended, then it is in the grounded extension. □

Because of the above two propositions, the agent can conclude to follow an optimal plan to achieve his goal. Besides, the notion of optimal plans is defined as the set of conclusions of ordinary arguments from the extensions, so the set of optimal plans becomes empty if an extension does not contain any ordinary arguments. The following proposition indicates the conditions for which the set of optimal plans is not empty.

Proposition 8. *Given a PAF = $(\mathcal{A}, \mathcal{D})$, Oplans$(PAF, sem) \neq \varnothing$ iff there exists an ordinary argument $\langle +v_a, \lambda_a \rangle$ such that it is not defeated by a blocking argument $\langle -v_b, \neg \lambda_b \rangle$ with $v_a < v_b$.*

Proof. Having $\text{Oplans}(PAF, sem) \neq \varnothing$ means that there is at least one extension which contains at least one ordinary argument. $\Rightarrow$: Suppose there does not exists an ordinary argument $\langle +v_a, \lambda_a \rangle$ such that it is not defeated by a blocking argument $\langle -v_b, \neg\lambda_b \rangle$ with $v_a \prec v_b$, which means that all the ordinary arguments (if exist) are defeated by a blocking argument and not self-defended against a blocking argument. In such a case, there exists a blocking argument that does not receive any defeats, which makes all the ordinary arguments rejected. Contradiction! $\Leftarrow$: If there exists an ordinary argument such that it is not defeated by a blocking argument or self-defended against a blocking argument, then (1) the ordinary argument does not receive any defeats and thus it should be contained in the grounded extension, or (2) the ordinary argument is in a cycle with a blocking argument and thus it should be contained in a preferred extension, or (3) the ordinary argument receives defeats from other ordinary arguments and thus there is always an ordinary argument accepted. Hence, $\text{Oplans}(PAF, sem)$ is not an empty set. $\qquad\square$

Example 5. *The plan-based argumentation framework PAF can be represented as Figure. 3. Because*

$$\mathcal{E}_{prf}(PAF) = \mathcal{E}_{grd}(PAF) = \mathcal{E}_{stb}(PAF) =$$
$$\{\{\langle +pv, (\alpha_2, \alpha_4, \alpha_5)\rangle, \langle +sf, (\alpha_2, \alpha_4, \alpha_5)\rangle,$$
$$\langle -pv, \neg(\alpha_1, \alpha_6)\rangle, \langle -sf, \neg(\alpha_2, \alpha_3)\rangle\}\}$$

and thus $\text{Oplans}(PAF, sem) = \{(\alpha_2, \alpha_4, \alpha_5)\}$, *the agent can follow plan* $(\alpha_2, \alpha_4, \alpha_5)$ *to get to a pharmacy.*

4. Related Work

Our approach is closely related to the value-based argumentation framework (VAF) [2][1]. The differences are as follows: firstly, their framework allows us to reason about what we should do given all the available actions, while our framework allows us to reason about what we should do given all the available plans each of which is a sequence of actions; secondly, in their framework informal arguments are constructed through asking critical questions associated with an argument scheme and a transition system, while in our framework agents construct formal arguments through checking formulas with respect to the underlying transition system; thirdly, the aim of VAF (having an audience as an element) is to explain different agents' choices based on their social values, while the aim of our paper is to design an approach for agents' planning. Existing work combines modal logic and argumentation in different ways. Proietti and Yuste-Ginel combines both techniques to reason about the knowledge of arguments in a debate and its dynamics [15], and Bulling etc. combine both techniques to reason about the abilities of coalitions of agents and the formation of coalitions [16]. Both use modal logic as meta-language to argumentation, while we use argumentation as meta-language to modal logic.

5. Conclusions

In this paper, we developed a logic-based framework that combines modal logic and argumentation for value-based practical reasoning. Modal logic is used as a technique to

represent and verify whether a plan with its local properties of value promotion or demotion can be followed to achieve an agent's goal. Seeing a verification result as an argument and defining a defeat relation based on an attack relation and preference over values, we then proposed an argumentation-based approach that allows an agent to reason about his plans using the verification results. Thus, our framework not only offers an approach for value-based practical reasoning with plans, but also makes a bridge between modal logic and argumentation in terms of argument construction. In the future, we would like to extend our framework by allowing an agent to have multiple goals instead of one goal as we assumed, or taking the actions of other agents into account in the context of multi-agent systems. More interestingly, we can study how autonomous agents are properly aligned with human values through adding constraints to the decision-making mechanism presented in this paper.

References

[1] Atkinson K, Bench-Capon T. Taking account of the actions of others in value-based reasoning. Artificial Intelligence. 2018;254:1-20.
[2] Bench-Capon T, Atkinson K, McBurney P. Using argumentation to model agent decision making in economic experiments. Autonomous Agents and Multi-Agent Systems. 2012;25(1):183-208.
[3] Liao B, Anderson M, Anderson SL. Representation, justification, and explanation in a value-driven agent: an argumentation-based approach. AI and Ethics. 2021;1(1):5-19.
[4] Ågotnes T, van der Hoek W, Wooldridge M. Normative system games. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems; 2007. p. 1-8.
[5] Knobbout M, Dastani M. Reasoning under compliance assumptions in normative multiagent systems. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems; 2012. p. 331-40.
[6] Alechina N, Dastani M, Logan B. Reasoning about Normative Update. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI '13. AAAI Press; 2013. p. 20–26.
[7] Luo J, Meyer JJ, Knobbout M. A formal framework for reasoning about opportunistic propensity in multi-agent systems. Autonomous Agents and Multi-Agent Systems. 2019;33(4):457-79.
[8] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence. 1995;77(2):321-57.
[9] Prakken H. Combining sceptical epistemic reasoning with credulous practical reasoning. COMMA. 2006;144:311-22.
[10] Amgoud L, Prade H. Formalizing practical reasoning under uncertainty: An argumentation-based approach. In: 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07). IEEE; 2007. p. 189-95.
[11] Knobbout M, Dastani M, Meyer JJ. A dynamic logic of norm change. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence; 2016. p. 886-94.
[12] Knobbout M, Dastani M, Meyer JJC. Reasoning about dynamic normative systems. In: European Workshop on Logics in Artificial Intelligence. Springer; 2014. p. 628-36.
[13] Modgil S, Prakken H. A general account of argumentation with preferences. Artificial Intelligence. 2013;195:361-97.
[14] Liao B, Oren N, van der Torre L, Villata S. Prioritized norms in formal argumentation. Journal of Logic and Computation. 2019;29(2):215-40.
[15] Proietti C, Yuste-Ginel A. Dynamic epistemic logics for abstract argumentation. Synthese. 2021;199(3):8641-700.
[16] Bulling N, Dix J, Chesnevar CI. Modelling coalitions: ATL+ argumentation. In: AAMAS (2); 2008. p. 681-8.