Normative Practical Reasoning via Argumentation and Dialogue

Submission XXX

Abstract

In a normative environment an agent’s actions are not only directed by its goals but also by the norms imposed on it. However, the potential conflicts within and between the agent’s goals and norms makes decision-making in these frameworks a challenging task. The questions we address in this paper are: (i) how should an agent act in a normative environment? (ii) how can the agent explain why it acted in a certain way? We propose a solution in which a normative planning problem serves as the basis for a practical reasoning approach based on argumentation. The properties of the best plan(s) with respect to goal achievement and norm compliance are mapped to arguments that are used to explain why a plan is justified, using an existing proof dialogue game.

1 Introduction

Autonomous agents operating in a dynamic environment must be able to reason about actions in pursuit of their goals. An additional consideration for such agents are the regulative norms imposed on them that define what an agent is obliged or forbidden to do. To avoid punishment, agents must comply with norms while pursuing their goals. However, if complying with a norm hinders a more important goal or norm, the agent should consider violating it. In order to make a decision about what to do, an agent therefore needs to generate all possible courses of actions (i.e., plans) and weigh up the importance of goal achievement and norm compliance against the cost of goals being ignored and norms being violated, in different plans. Although practical reasoning frameworks that take norms into account exist (e.g., [Broersen et al., 2001; Kollingbaum and Norman, 2003]), little attention has been paid to the explanation of the agents’ decision making in such frameworks. Such explanation is important in several contexts, including human-agent teams and debugging agents. To address this shortcoming, we therefore propose conducting practical reasoning using argumentation.

Argumentation is a discipline that has dealt with issues of handling inconsistency and decision-making [Dung, 1995; Amgoud and Prade, 2009] for a long time. In addition, the dialogical aspect of argumentation makes it an appropriate tool to generate explanation for a decision made using this technique (e.g. [Fan and Toni, 2015; Caminada et al., 2014b]). Although argumentation has been extensively used in practical reasoning (e.g., [Rahwan and Amgoud, 2006; Atkinson and Bench-Capon, 2007]), integrating the reasoning and dialogical aspect of argumentation for decision-making and its explanation is not addressed in existing approaches.

In this paper we propose an argumentation-based approach to normative practical reasoning that uses dialogue games to provide an intuitive overview of agent’s reasoning. In achieving this aim, the following contributions are made: (i) we formalise a set of argument schemes and critical questions [Walton, 1996] that aim at checking the justifiability of plans with respect to goal satisfaction and norm compliance/violation; (ii) we offer a novel decision criterion that identifies the best plan(s) both in presence and absence of preferences over goals and norms; and (iii) we investigate the properties of the best plan(s) and propose a concrete application for the recently developed preferred dialogue games [Caminada et al., 2014a] that uses these properties to generate an explanation for the justifiability of the best plan(s).

2 Model

This section introduces a model for normative practical reasoning based on STRIPS planning [Fikes and Nilsson, 1971].

Definition 1 (Normative Planning Problem). A normative planning problem is a tuple $P = (FL, \Delta, A, G, N)$ where $FL$ is a set of fluents; $\Delta \subseteq FL$ is the initial state; $A$ is a finite, non-empty set of durative actions; $G$ is the set of agent goals; $N$ is a set of action-based norms imposed on the agent.

Fluents $FL$ is a set of domain fluents. A literal $l$ is a fluent or its negation. For a set of literals $L$, we define $L^+ = \{fl \text{ s.t. } fl \in L\}$ and $L^- = \{fl \text{ s.t. } \neg fl \in L\}$. $L$ is well-defined if $L^+ \cap L^- = \emptyset$. A state $s \subseteq FL$ is determined by those fluents true at a given time, other fluents are considered false. A state $s$ satisfies literal $fl$, denoted as $s \models fl$, if $fl \in s$, and satisfies literal $\neg fl$, denoted $s \models \neg fl$, if $fl \notin s$.

Actions An action $a = (pr, ps, d)$ is composed of well-defined sets of literals $pr(a)$, $ps(a)$ that represent $a$’s pre- and postconditions and a number $d(a) \in \mathbb{N}$ representing the action’s duration. Postconditions are divided into a set of add postconditions $ps(a)^+$ and a set of delete postconditions $ps(a)^-$.
An action $a$ can be executed in a state $s$ if its preconditions hold in that state. The postconditions of a durative action are applied in the state $s$ at which the action ends, by adding the positive postconditions belonging to $ps(a)^+$ and deleting the negative postconditions belonging to $ps(a)^-$. 

**Goals Achievement** goals need to instantaneously achieve a certain state of affairs. Each $g \in G$ is a well-defined set of literals $g = \{r_1, \ldots, r_n\}$, known as goal requirements (denoted as $r_i$), that should hold in order to satisfy the goal.

**NORMS** An action-based norm is defined as a tuple $n = (d, \sigma, a_{con}, a_{sub}, dl)$, where $d, \sigma \in \{o, f\}$ is the deontic operator denoting obligation or prohibition; $a_{con} \in A$ is the action that activates the norm; $a_{sub} \in A$ is the action that is the subject of the obligation or prohibition; and $dl \in \mathbb{N}$ is the norm deadline relative to the execution of the action $a_{con}$, that is the activation condition of the norm.

### 2.1 Semantics

Let $P = (\mathcal{FL}, \Delta, A, G, N)$ be a normative planning problem as described previously. Also let $\pi = (\langle a_0, 0 \rangle, \ldots, \langle a_i, t_a \rangle)$ with $a_i \in A$ and $t_a \in \mathbb{Z}^+$ be a sequence of actions $a_i$ executed at time $t_a$, s.t. $\forall i < j, t_a < t_b$. The total duration of a sequence of actions is calculated as follows: $Makespan(\pi) = \max(t_a + d(a_i))$. The execution of a sequence of actions from a given starting state $s_0 = \Delta$ brings about a sequence of states $S(\pi) = \langle s_0, \ldots, s_m \rangle$ for every discrete time interval from $0$ to $m$, where $m = Makespan(\pi)$. The transition relation between two states is given in Equation (1) below. If an action $a_i$ ends at time $k$, state $s_k$ results from removing all delete postconditions and adding all add postconditions of action $a_i$ to state $s_{k-1}$. Thus, $\forall 0 < k \leq m :$

$$s_k = \begin{cases} (s_{k-1} \setminus ps(a_i)^-) \cup ps(a_i)^+ & k = t_a + d(a_i) \\ s_{k-1} & \text{otherwise} \end{cases}$$

(1)

**satisfies a goal** if there is a state that satisfies the goal: $\pi \models g$ if $\exists s_k \in S(\pi)$ s.t. $s_k \models g$. The set of satisfied goals by $\pi$ is denoted as $G_\pi$.

**complies with an obligation** if the action that is the subject of the obligation, $a_{sub}$, occurs during the compliance period (i.e. between when the condition holds and when the deadline expires):

$$\pi \models n \iff \langle a_{con}, t_{a_{con}} \rangle, (a_{sub}, t_{a_{sub}}) \in \pi \text{ s.t. } t_{a_{sub}} \subseteq \langle t_{a_{con}} + d(a_{con}), dl + t_{a_{con}} + d(a_{con}) \rangle$$

If $a_{sub}$ does not occur during the compliance period, the obligation is violated: $\pi \not\models n$.

**complies with a prohibition** if the prohibition’s subject action $a_{sub}$ does not occur during the compliance period:

$$\pi \models n \iff \langle a_{con}, t_{a_{con}} \rangle \in \pi, \exists \langle a_{sub}, t_{a_{sub}} \rangle \in \pi \text{ s.t. } t_{a_{sub}} \subseteq \langle t_{a_{con}} + d(a_{con}), dl + t_{a_{con}} + d(a_{con}) \rangle$$

If $a_{sub}$ occurs during the compliance period, the prohibition norm is violated: $\pi \not\models n$.

We assume that the norm deadlines end before $m = Makespan(\pi)$. Therefore, all the activated norms in $\pi$, denoted as $N_\pi$, are either complied with or violated by time $m$.

### 2.2 Conflict

In this section different types of conflicts are discussed; and it is defined which sequence of actions are identified as a plan w.r.t these conflicts. We consider a running example where an agent has the goals of going on strike, submitting a report and getting a certificate of some sort. However, if the agent goes on maternity leave, it cannot go to the office and submit the report. Moreover, if the agent goes on strike, it cannot go to office or attend any meeting.

**Definition 2** (Conflicting Actions). Actions $a_i$ and $a_j$ have a concurrency conflict if the preconditions or postconditions of $a_i$ contradict the preconditions or postconditions of $a_j$.

$$cf_{\text{action}} = \{(a_i, a_j) \text{ s.t. } \exists r \in pr(a_i) \cup ps(a_i), \neg r \in pr(a_j) \cup ps(a_j)\}$$

**Definition 3** (Conflicting Goals). Goal $g_i$ and $g_j$ are trivially in conflict iff satisfying them requires bringing about conflicting state of affairs.

$$cf_{\text{goal}} = \{(g_i, g_j) \text{ s.t. } \exists r \in g_i, \neg r \in g_j\}$$

**Example 1.** strike $= \{\text{union member, } \neg \text{at office, } \neg \text{meeting attended}\}$ and submission $= \{\text{at office, report finalised}\}$ are conflicting.

**Definition 4** (Conflicting Obligations and Goals). $n = (a_{con}, a_{sub}, dl)$ and $g$ are trivially in conflict, if executing action $a_{sub}$ that is the subject of the obligation, brings about postconditions that are in conflict with the requirements of $g$.

$$cf_{\text{goalobl}} = \{(g, n) \text{ s.t. } \exists r \in g, \neg r \in ps(a_{sub})\}$$

**Example 2.** strike and $n_1 = (a, \text{get company funding, attend meeting, 2})$, where the agent is obliged to attend a meeting on behalf of the company if it uses the company funding, are in conflict, since the postconditions of attend meeting prevents the agent from going on strike.

**Definition 5** (Conflicting Prohibitions and Goals). A prohibition norm $n = (f, a_{con}, a_{sub}, dl)$ and a goal $g$ are trivially in conflict, if the postconditions of $a_{sub}$ contribute to satisfying $g$, but executing action $a_{sub}$ is prohibited by norm $n$.

$$cf_{\text{goalpro}} = \{(g, n) \text{ s.t. } \exists r \in g, r \in ps(a_{sub})\}$$

**Example 3.** submission $= \{\text{at office, report finalised}\}$ and $n_2 = (f, \text{take maternity leave, go to office, 6})$ are in conflict since taking maternity leave prevents the agent from going to the office and hence prevents fulfilling the goal of submission: submission, $n_2 \in cf_{\text{goalnorm}}$.

The entire set of conflicting goals and norms is defined as:

$$cf_{\text{goalnorm}} = cf_{\text{goalobl}} \cup cf_{\text{goalpro}}$$

**Definition 6** (Conflicting Obligations). $n_1 = (a, a_{con}, a_{sub}, dl)$ and $n_2 = (b, b_{con}, b_{sub}, dl')$ are in conflict in the context of $\pi$ if the obligations in $n_1$, i.e. $a_{sub}$, and $n_2$, i.e. $b_{sub}$ have a concurrency conflict; and action $a_{sub}$ is in progress during the entire period over which the agent is obliged to execute action $b_{sub}$.

$$cf_{\text{oblobl}} = \{(n_1, n_2) \text{ s.t. } (a_{con}, t_{a_{con}}), (b_{con}, t_{b_{con}}) \in \pi; (a_{con}, b_{con}) \in cf_{\text{action}}; t_{a_{sub}} \subseteq \langle t_{a_{con}} + d(a_{con}), t_{a_{con}} + d(a_{con}) + dl \rangle; t_{b_{sub}} \subseteq \langle t_{b_{con}} + d(b_{con}), t_{b_{con}} + d(b_{con}) + dl' \rangle \}$$

**Example 4.** Due to the concurrency conflict between actions attend meeting and attend interview, in $n_1 = (a, \text{get company funding, attend meeting, 2})$ and $n_4 = (b,
theory meeting, and funding (Plan) in Section 3.2. For-
impossible for the agent to satisfy all its goals while com-
deal with the issue of choosing the best plan. The obligation to, and a pro-
bids the agent to execute action $a_{sub}$ during the entire period over which obligation $n_1$ obliges the agent to take $a_{sub}$.

$$c_{oblpro} = \{(n_1, n_2) \mid (a_{con}, t_{con}), (b_{con}, t_{con}) \in \pi; \ | t_{con} + d(a_{con}), t_{con} + d(a_{con}) + d(\lambda) \rangle \subseteq \langle t_{b_{con}} + d(b_{con}), t_{b_{con}} + d(b_{con}) + d(\lambda') \rangle \}$$

**Example 5.** The obligation to, and a prohibition from attend meeting in $n_1 = \langle o, get\_company\_funding, attend\_meeting, 2 \rangle$ and $n_3 = \langle f, take\_maternity\_leave, attend\_meeting, 6 \rangle$ can cause a normative conflict in some $\pi$: $(n_1, n_3) \in c_{oblpro}^\pi$.

All together, two sets $c_{oblpro}^\pi$ and $c_{oblpro}^\pi$ constitute the set of conflicting norms: $c_{oblpro}^\pi = c_{oblpro}^\pi \cup c_{oblpro}^\pi$.

**Definition 7 (Plan).** A sequence of actions $\pi = ((a_0, 0), \ldots, (a_n, t_{an}))$ is a plan for the normative planning problem $P = (FL, \Delta, A, G, N)$ iff the following conditions hold:

- The fluent values in $\Delta$ (and nothing else) hold in the initial state: $s_0 = \Delta$
- The preconditions of action $a_i$ hold at time $t_{ai}$ and throughout the execution of $a_i$:
  $$\forall k \in [t_{ai}, t_{ai} + d(a_i)], s_k = pr(a_i)$$
- The set of goals satisfied by plan $\pi$ is a non-empty ($G_\pi \neq \emptyset$) consistent subset of goals:
  $$G_\pi \subseteq G \text{ and } \not\exists g_i, g_j \in G_\pi \text{ s.t. } (g_i, g_j) \in c_{goal}$$
- There is no concurrency conflict between actions that are executed concurrently:
  $$\not\exists (a_i, t_{ai}), (a_j, t_{aj}) \in \pi \text{ s.t. } t_{ai} \leq t_{aj} < t_{ai} + d(a_i), (a_i, a_j) \in c_{action}$$
- There is no conflict between norms complied with:
  $$\not\exists n_i, n_j \in N_{cmp(\pi)} \text{ s.t. } (n_i, n_j) \in c_{norm}$$
- There is no conflict between goals satisfied and norms complied with:
  $$\not\exists g \in G_\pi \text{ and } n \in N_{cmp(\pi)} \text{ s.t. } (g, n) \in c_{goalnorm}$$

Having defined the set of plans, $\Pi$, in the next section we deal with the issue of choosing the best plan.

## 3 Identifying the Best Plan

The conflict between agent’s goals and norms often makes it impossible for the agent to satisfy all its goals while complying with all norms triggered in a plan. In this section we show how to treat each plan as a proposal of actions and how to use argumentation schemes to check the justifiability of a plan proposal with respect to conflicts and preferences, as a step toward identifying the best plan(s) in Section 3.2.

### 3.1 Argumentation Framework

An argumentation framework (AF) consists of a set of arguments and attacks between them [Dung, 1995]: $AF = (Arg, Att)$, $Att \subseteq Arg \times Arg$. In scheme-based approaches [Walton, 1996] arguments are expressed in natural language and a set of critical questions is associated with each scheme, identifying how the scheme can be attacked. Below, we introduce a set of argument schemes and critical questions to reason about a plan proposal w.r.t. goals it satisfies and norms it complies with or violates.

**Definition 9 (Plan Argument $Arg_\pi$).** A plan argument claims that the agent should execute a proposed sequence of actions because that leads to satisfying a set of goals, and complying with a set of norms, although it violates some norms:

- In the initial state $\Delta$
- The agent should execute sequence of actions $\pi$
- Which will satisfy set of goals $G_\pi$ and complies with set of norms $N_{cmp(\pi)}$ and violates set of norms $N_{vol(\pi)}$

**Definition 10 (Goal Argument $Arg_g$).** A goal argument claims that a feasible goal should be satisfied:

- Goal $g$ is a feasible goal of the agent
- Therefore, satisfying $g$ is required

The set of goal argument for a plan is denoted as $Arg_G$.

**Definition 11 (Norm Argument $Arg_n$).** A norm argument claims that an activated norm should be complied with:

- $n$ is an activated norm imposed on the agent in plan $\pi$
- Therefore, complying with $n$ is required in $\pi$

The set of norm argument for a plan is denoted as $Arg_{G_n}$.

### Critical Questions Associated with Plan Scheme

**CQ1:** Is there any attack from a goal argument to $Arg_\pi$? This CQ results in an undercut attack (symmetric by definition) from a goal argument to a plan argument, when the goal is not satisfied in the plan:

$$\forall Arg_g \in Arg_g \text{ if } \pi \neq g \text{ then } (Arg_g, Arg_\pi) \in Att$$

**CQ2:** Is there any attack from a norm argument to $Arg_\pi$? This CQ results in an undercut from a norm argument to a plan argument, when the norm is violated in the plan:

$$\forall Arg_n \in Arg_{G_n} \text{ if } \pi \neq n \text{ then } (Arg_n, Arg_\pi) \in Att$$

### Critical Questions Associated with Goal Scheme

**CQ3:** What goal arguments might attack $Arg_g$? This CQ results in a rebutt attack (symmetric by definition) between arguments for conflicting goals:

$$\forall Arg_g, Arg_{g'} \in Arg_G \text{ if } (g, g') \in c_{goal} \text{ then } (Arg_g, Arg_{g'}) \in Att$$

**CQ4:** What norm arguments might attack $Arg_g$? This CQ results in a rebutt attack between arguments for a goal and a norm that are in conflict:

$$\forall Arg_g \in Arg_G, Arg_n \in Arg_{G_n} \text{ if } (g, n) \in c_{goalnorm} \text{ then } (Arg_g, Arg_n) \in Att$$

### Critical Questions Associated with Norm Scheme

**CQ4:** What goal arguments might attack the norm presented by $Arg_n$? The previous critical question, is associated with argument schemes for norms as well as goals, hence the repetition of the number of critical question.

$$\forall Arg_g \in Arg_G, Arg_n \in Arg_{G_n} \text{ if } (n, g) \in c_{goalnorm} \text{ then } (Arg_n, Arg_g) \in Att$$

---

1 A goal is feasible if there is at least one plan that satisfies it.
CQ5: What norm arguments might attack the norm presented by Argn? Conflict between two norms is defined as a contextual conflict that depends upon the context of the plan in which the norms are activated.

\[ \forall \text{Arg} \alpha, \text{Arg} \beta \in \text{Arg} \text{N}_\gamma \]

if \((n,n') \in \text{cf}^n_{\text{norm}} \), then \((\text{Arg} \alpha, \text{Arg} \beta) \in \text{Att} \)

Preferences between arguments distinguish an attack from a defeat (i.e., a successful attack [Amgoud and Cayrol, 2002]). The attack from one argument to another is a defeat if the latter argument is not preferred over the former. However, as discussed in [Prakken, 2012], rebuttal attacks are preference-dependent, whereas undercut arguments are preference-independent. Thus, attacks due to CQ3, CQ4 and CQ5 need to be resolved, while attacks caused by CQ1 and CQ2 are preference independent, always resulting in defeat.

We define \( \geq^g \) as a partial preorder on \( G \cup \text{N} \). Symbol \( \geq^g \) denotes the strict relation corresponding to \( \geq^g \). Also, \((\alpha, \beta) \in \sim^g \) iff \((\alpha, \beta) \in \geq^g \) and \((\beta, \alpha) \in \geq^g \). The preferences between the goal and norm arguments result from the preference relation between these entities: \((\text{Arg} \alpha, \text{Arg} \beta) \in \geq \) iff \((\alpha, \beta) \in \geq^g \).

An AF for a plan proposal consists of the argument for the plan itself, a set of arguments for goals and arguments for norms that are activated in that plan. Although the set of goal arguments in AFs for plan proposals remain the same across the AFs, the set of norm arguments differs from one to another depending on the norms that are activated in each.

Definition 12 (Plan Proposal AF). The AF for plan proposal \( \pi \) is \( \text{AF}_\pi = \langle \text{Arg}, \text{Def} \rangle \), where \( \text{Arg} = \text{Arg} \gamma \cup \text{Arg} G \cup \text{Arg} \text{N}_\gamma \) and \( \text{Def} \) is defined as: \( \forall \text{Arg} \alpha, \text{Arg} \beta \in \text{Arg} \), \((\text{Arg} \alpha, \text{Arg} \beta) \in \text{Def} \) iff \((\text{Arg} \alpha, \text{Arg} \beta) \in \text{Att}_{\text{CQ1-5}} \) and \((\text{Arg} \alpha, \text{Arg} \beta) \notin \geq \).

The next section explains how an AF for a plan proposal is evaluated and used toward identifying the best plan(s).

3.2 Evaluating the Argumentation Framework

Argumentation semantics are a means for evaluating arguments in an AF and various semantics have been introduced since the proposal of Dung’s AF [Dung, 1995]. Among these semantics, preferred is repeatedly proposed [Caminada, 2006; Prakken, 2006; Oren, 2013] to reason about and toward actions. Caminada [2006] provides an intuitive way to identify the status of arguments w.r.t. various semantics through labellings. Here, an argument is respectively, labelled in, out and undec, if it is acceptable, rejected and undecided under a certain semantics. In a complete labelling, an argument is labelled in iff all its attackers are labelled out, and the argument is labelled out iff there exists an attacker for it that is labelled in. A complete labelling in which the set of arguments labelled in are maximal (w.r.t. set inclusion) is a preferred labelling. Intuitively, an argument is credulously accepted under preferred semantics if it is labelled in by at least one preferred labelling.

Definition 13 (Justified Plans). Plan \( \pi \) is justified if \( \text{Arg} \pi \) is labelled in by at least one preferred labelling for \( \text{AF}_\pi \): \( \exists \text{E}_{pr} \text{ s.t. } \text{Arg} \pi \in \text{in}(\text{E}) \).

Although all justified plans are internally consistent, they can still be disagreed with externally for different reasons.

That is, there might be further criteria to take into account when identifying the best plan among justified plans. We define the criteria for the best plan(s) using an established ordering principle in argumentation, the Democratic principle: \((S_i, S_j) \in \geq \) iff \( \forall \beta \in S_j \setminus S_i, \exists \alpha \in S_i \setminus S_j \) s.t. \((\alpha, \beta) \in \geq \).

Since the preferences over goals and norms is partial, comparing two plans based on the set of goals and norms is not always possible. Therefore, absent such preference information, the best plan(s) must satisfy the best goal while violating the fewest norms. We start by defining the goal-dominant and norm-dominant plans, based on which a better than relation between plans is defined.

Definition 14 (Goal-dominance). Plan \( \pi_i \) goal-dominates \( \pi_j \) denoted as \((\pi_i, \pi_j) \in \geq \) if:

1. \( (G_{\pi_i}, G_{\pi_j}) \in \geq \) or \( \text{Else} \).
2. \( |G_{\pi_i}| \geq |G_{\pi_j}| \).

Definition 15 (Norm-dominance). Plan \( \pi_i \) norm-dominates \( \pi_j \) denoted as \((\pi_i, \pi_j) \in \geq \) if:

1. \( (N_{\text{vol}}(\pi_i), N_{\text{vol}}(\pi_j)) \in \geq \) or \( \text{Else} \).
2. \( |N_{\text{vol}}(\pi_i)| \geq |N_{\text{vol}}(\pi_j)| \).

It is straightforward to show that \( \geq_G \) and \( \geq_N \) are total preorders on a set of plans \( \Pi \).

Definition 16 (Plan Comparison). Plan \( \pi_i \) is better than \( \pi_j \), denoted as \((\pi_i, \pi_j) \in \pi \) if:

1. \( \pi_i \) is justified and \( \pi_j \) is not; or
2. \( \pi_i \) and \( \pi_j \) are both justified and \((\pi_i, \pi_j) \in \geq \) or \( \text{Else} \).

Plan \( \pi_i \) is as good as \( \pi_j \), denoted as \((\pi_i, \pi_j) \in \pi \) if \((\pi_i, \pi_j) \not\in \pi \) or \((\pi_j, \pi_i) \not\in \pi \).

The relation \( \pi \) is irreflexive, asymmetric and transitive, while \( \sim_\pi \) is an equivalence relation on \( \Pi \).

Definition 17 (Equivalence Classes). Given \( \pi \in \Pi \), let \( [\pi]_{\sim} \) denote the equivalence class to which \( \pi_i \) belongs. \((\pi_i, \pi_j) \in \pi \) iff \( \pi_i = \pi_j \) or \((\pi_i, \pi_j) \in \sim_\pi \).

Definition 18 (Best Plan(s)). Plan \( \pi_i \) is (one of) the best plan(s) for the agent to execute iff

- \( \pi_i \) is justified, and
- \( \forall \pi_j \) such that \((\pi_i, \pi_j) \in \pi \).

Example 6. Assume an agent with three goals strike, submission (Example 1), and certificate = \{course_fee_paid, theory_test_done, interviewed\} and four norms n1, n2, n3, and n4 (Examples 2, 3, 4, and 5). The agent prefers satisfying goal submission to complying with norm n2: submission \( \geq n2 \), also it prefers complying with n4 rather than n1: n4 \( \geq n1 \). Let \( \pi_1, \pi_2, \pi_3, \pi_4 \in \Pi \): 

- \( \pi_1 \models \text{submission}, \text{N}_{\text{active}(\pi_1)} = \{n1\} \), \( \pi_1 \models n1 \)
- \( \pi_2 \models \text{submission}, \text{certificate}, \text{N}_{\text{active}(\pi_2)} = \{n1, n4\} \), \( \pi_2 \models n1, \pi_2 \models n4 \)
- \( \pi_3 \models \text{submission}, \text{certificate}, \text{N}_{\text{active}(\pi_3)} = \{n1, n2, n3, n4\} \), \( \pi_3 \models n3, n4 \), \( \pi_3 \not\models n1, n2 \)
- \( \pi_4 \models \text{strike, certificate, N}_{\text{active}(\pi_4)} = \{n1, n2, n3, n4\} \), \( \pi_4 \models n2, n3, n4 \), \( \pi_4 \not\models n1 \).
Figure 1 displays the argumentation graph associated with each of these plans. Plan π1 is not justified, whereas π2, π3 and π4 all are. Thus, the first condition in Definition 18 holds for the last three plans. Since the preferences provided over goals and norms is minimal, in this example the number of goals satisfied and norms violated determines the best plans as follows: although |G(π2)| = |G(π3)| = |G(π1)|, |N_{vol}(π2)| = |N_{vol}(π3)| < |N_{vol}(π1)|. Therefore, π2 > π3, π4 > π3, and π2 ∼ π4, which makes π2 and π4 the best plans.

3.3 Properties

First, we confirm the satisfaction of rationality postulates [Caminada and Amgoud, 2007]. Second, we investigate the properties of the best plan(s) and the preferred extensions that include it.

Property 1. Closure: The conclusions of any extension (in labelled arguments) are closed under strict rules.

Proof. Plan, goal and norm arguments are built based on defeasible rules (schemes). With no strict rules the property follows immediately.

Property 2. Direct Consistency: The conclusions of any extension are consistent.

Proof. Suppose the conclusions of the extension E are inconsistent, i.e., there are arguments Argα, Argβ ∈ E such that:
- Argα’s conclusion requires executing plan π and Argβ’s conclusion requires satisfying goal γ with norm n, while γ is not satisfied/n is violated in π. Thus, Argβ defeats Argα; E is not conflict-free and cannot be an extension.
- Argα’s conclusion requires satisfying goal γ with norm n and Argβ’s conclusion requires satisfying goal γ with norm n’, while γn and γn’ are inconsistent. Thus, Argα attacks Argβ and vice versa. Due to the preferences, at least one of these attacks is identified as defeat and therefore E is not conflict-free and not an extension.

Property 3. Indirect Consistency: The closure under strict rules of the conclusions of any extension is consistent.

Proof. With no strict rules the property follows immediately.

Property 4. If a plan argument is labelled in by preferred labelling L, the arguments representing all the goals that it does not satisfy and norms it violates are labelled out by L and vice versa:
Argg ∈ in(L) ↔ Argg ∈ G \ Gπ & Argn ∈ N_{vol}(π) ≝ out(L).

Proof. Every preferred labelling is a complete labelling. An argument is labelled in by a complete labelling if all its attackers are labelled out. Therefore, a plan argument is labelled in by a preferred labelling iff all its attackers, namely the arguments for goals that it does not satisfy and norms that it violates, are labelled out by that labelling.

Property 5. If a plan argument is labelled in by preferred labelling L, the arguments representing all the goals that it satisfies and norms it complies with are also labelled in:
Argg ∈ in(L) ⇒ Argg ∈ Gπ & Argn ∈ N_{cmp}(π) ∈ in(L).

Proof. Since Argg ∈ in(L), from Property 4 we know that Argg ∈ Gπ ∪ Argn ∈ N_{vol}(π) ≝ out(L). We also know from the definition of a plan that Argg ∈ Gπ & Argn ∈ N_{cmp}(π) is conflict free. Since all possible attackers of Argg ∈ Gπ & Argn ∈ N_{vol}(π) belong to Argg ∈ Gπ & Argn ∈ N_{vol}(π) and Argg ∈ Gπ & Argn ∈ N_{vol}(π) are all labelled out, we conclude that Argg ∈ Gπ & Argn ∈ N_{cmp}(π) ∈ in(L).

Note that from Argg ∈ Gπ & Argn ∈ N_{cmp}(π) ∈ in(L) one cannot conclude that Argg ∈ in(L), as there might be justified goals or norms not satisfied/complied with in the plan.

Property 6. There is no more than one preferred labelling in which Argg ∈ in(L).

Proof. From Property 4 and 5 we know that if Argg ∈ in(L) then Argg ∈ Gπ ∪ Argn ∈ N_{vol}(π) ≝ out(L) and Argg ∈ Gπ & Argn ∈ N_{cmp}(π) ∈ in(L). Since every preferred labelling is a complete labelling and the following property holds for complete labellings: if out(Lcmp1) = out(Lcmp2) then Lcmp1 = Lcmp2; we conclude that there is no more than one preferred labelling in which Argg ∈ in(L).

Property 7. If Argg ∈ in(L), L is a stable labelling.

Proof. In Property 4 we showed that if Argg ∈ in(L) then Argg ∈ Gπ ∪ Argn ∈ N_{vol}(π) ≝ out(L) and Argg ∈ Gπ & Argn ∈ N_{cmp}(π) ∈ in(L), which makes the undec(L) = ∅. A preferred labelling with undec(L) = ∅ is a stable labelling. Therefore, L is a stable labelling.

Property 8. Let ≥^n be a total preorder on G ∪ N and therefore ≥ be a total preorder on goal and norm arguments. If Argg ∈ in(L), and the set of arguments for the most preferred goals and norms, Pref(Argg), is conflict free, all arguments belong to Pref(Argg) are labelled in by L.

Proof. Elements of set Pref(Argg) cannot be defeated, since the set itself is conflict-free and the rest of arguments belong to Argg \ Pref(Argg) cannot defeat elements of Pref(Argg), since that implies an attack from a less preferred argument to a more preferred one has resulted in a defeat, which is contrary to assumption. Assume that ∃Argg ∈ Pref(Argg) such that Argg /∈ in(L). If ∃Argg /∈ in(L) s.t. (Argg, Argg) ∈ Def then Argg should have been labelled in by L otherwise it is contrary to the assumption of maximality of preferred labellings. If ∃Argg /∈ in(L) s.t. (Argg, Argg) ∈ Def then Argg /∈ in(L) s.t. (Argg, Argg) ∈ Def, which is contradictory to the fact that Argg cannot be defeated. Therefore, all arguments Pref(Argg) are labelled in by in(L).

4 Explaining the Justifiability of the Best Plan

In this section we exploit an existing dialogue for preferred semantics known as Socratic Discussion [Caminada et al., 2014a] to provide an explanation for the justifiability of the best plan(s). Deciding if an argument is in at least one preferred extension amounts to deciding if it is at least in one admissible extension (i.e. it is labelled in by at least one admissible labelling). In an admissible labelling if an argument is labelled in, all attackers are labelled out, and if an argument is labelled out, it has an attacker that is labelled in.
Definition 19 (Socratic Discussion [Caminada et al., 2014a]). Let \( AF = \langle Arg, Def \rangle \). The sequence of moves \( [\Delta_1, \Delta_2, \ldots, \Delta_n] \) \(( n \geq 1)\) is a Socratic discussion iff: (i) each odd move (M-move) is an argument labelled in; (ii) each even move (S-move) is an argument labelled out; (iii) each argument moved by \( S \) attacks an argument moved by \( M \) earlier in the dialogue; (iv) each argument moved by \( M \) attacks an argument moved by \( S \) in the previous step; (v) S-moves cannot be repeated. Player \( S \) wins the discussion if there is an M-move and an S-move containing the same argument. Otherwise, the winner is the player that makes the last move.

Given that the agent’s best plans(s) \( \pi \) is labelled \( in \) by at least one preferred labelling, player \( M \) is guaranteed a winning strategy in a Socratic discussion with \( \Delta_1 = in(\text{Arg}_M) \). The even moves in the rest of dialogue are arguments labelled \( out \), which according to Property 4 are goals not satisfied or norms violated in \( \pi \). On the other hand, the rest of odd moves in the dialogue are arguments labelled \( in \), which according to Property 5 are goals satisfied or norms complied with in \( \pi \). Since each odd move attacks the even move in the previous step, during a dialogue the agent is able dialectically to explain why it did not satisfy a goal or violate a norm, which are the two causes of attacks to the plan proposals.

Example 7. This example shows a Socratic discussion \( \Delta = \langle \text{in(Arg}_{\pi_1}\rangle, \text{out(Arg}_{\text{sub}}\rangle, \text{in(Arg}_{\text{st}}\rangle, \text{out(Arg}_{n_1}\rangle, \text{in(Arg}_{n_4}\rangle \rangle \) for plan \( \pi_4 \).

- M: Plan \( \pi_4 \) is (one of) the best plan(s) and is justifiable.
- S: Why does the plan not satisfy goal submission?
- M: Because the plan satisfies goal strike that attacks goal submission.
- S: Why does the plan violate norm \( n_1 \) ?
- M: Because the plan satisfies norm \( n_4 \) that attacks norm \( n_1 \).

5 Related Work

One of the most well-known scheme-based approach in practical reasoning is [Atkinson and Bench-Capon, 2007]. Recently, [Oren, 2013] has proposed a similar scheme-based approach for normative practical reasoning, but unlike [Atkinson and Bench-Capon, 2007] arguments are constructed for a sequence of actions rather than every single action. As a result the schemes are simpler. Similar to the latter approach, in this paper, arguments are constructed for plans rather than actions. [Oren, 2013] assumes that the conflicts within and between goals and norms are inferred from paths, rather than being formulated at the formal model level. Thus, although it is possible to explain why a path is more preferred over another one, it is not possible to underpin why a path does not satisfy a goal or violate a norm. In contrast, we explicitly concern ourselves with why the agent does not satisfy a goal or violate a norm. In addition, in this work the explanation of justifiability of why a plan is (one of) the best plan(s) for the agent to execute is formulated using a dialogue game for preferred semantics.

There are few applications of dialogue games that use the dialogues for explanation purposes [Zhong et al., 2014; Fan and Toni, 2015; Caminada et al., 2014b]. In [Zhong et al., 2014] and [Fan and Toni, 2015] admissible dispute trees developed for Assumption-based Argumentation [Dung et al., 2009] are used to provide explanation for why a certain decision is better than another one. In [Caminada et al., 2014b] a dialogical proof procedure based on the grounded semantics dialogue game [Caminada and Podlaszewski, 2012] is created to justify the actions executed in a plan. Despite the popularity of the preferred semantics, they have not been used in applications in the past. Our work proposes a concrete application of preferred dialogue games in a practical reasoning domain.

6 Conclusion and Future Work

Argumentation has been used to study practical reasoning and decision-making in the past [Zhong et al., 2014; Rahwan and Angou; 2006; Amgoud and Prade, 2009; Oren, 2013]. In contrast to existing approaches, we propose a framework that integrates the reasoning and dialogical aspects of argumentation to undertake normative practical reasoning. The question of how should an agent act in a normative environment while it has conflicting goals and norm is answered using argumentation-based reasoning. Moreover, the question of how can the agent generate explanation for why it acted a certain way, is answered using an argumentation-based dialogue.

In the current approach the conflict within goals and between goals and norms is addressed trivially. Similar to conflicts between norms, these two types of conflicts can be addressed temporally, hence enriching the conflict detection. This is left for future work. Another interesting direction of future work is testing empirically if providing explanation, in particular in natural language, in practical reasoning domain raises the likelihood of human users accepting the recommendation of the system regarding the best plan(s).

References

[Amgoud and Cayrol, 2002] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of


