Temporal Accommodation of Legal Argumentation

Régis Riveret
Independent Researcher
France
regis.riveret@gmail.com

Antonino Rotolo
CIRSFID
University of Bologna
Bologna, Italy
antonino.rotolo@unibo.it

Giuseppe Contissa
Dept. of Law
European University Institute
Florence, Italy
giuseppe.contissa@eui.eu

Giovanni Sartor
Dept. of Law
European University Institute
Florence, Italy
giovanni.sartor@eui.eu

Wamberto Vasconcelos
Dept. of Computing Science
University of Aberdeen
Aberdeen, UK
wvasconcelos@acm.org

ABSTRACT

This paper proposes to integrate an argumentation framework with techniques from Temporal Constraint Satisfaction. Temporal constraints are thus embedded into legal argumentation to account for temporal aspects of legal reasoning. Through the accommodation of temporal constraints, the validity of arguments and of their conclusions is made relative to the time-points when the applied norms are alive, according to the adopted temporal perspective. A fixed-point semantics and an associated dialogue game are given.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: General; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Legal reasoning, Argumentation, Constraints.

Keywords

Legal reasoning, Argumentation, Constraints.

1. INTRODUCTION

The concept of time is of paramount importance in the law. All legal norms have effects that hold during certain times, though this takes place in different ways. Certain norms have effects persisting over time, until some subsequent events terminate such effects. For example, the provision "If one causes damage, one has to provide compensation" will generate an obligation to pay that will persist until the duty holder fulfills this obligation.

Other norms have effects holding at specific times if the conditions of these norms hold at such times. For example, the provision "If one is in a public building, one is forbidden to smoke" states that, if one is in a public building at a certain time, then one has the prohibition to smoke at that time, but this obligation will persist only as long as the duty holder is in a public building.

Temporal aspects are the specific concern of certain kinds of norms. For instance, some norms state deadlines (*e.g.*, the rule "The payment terms shall be in full upon receipt of invoice. Interest shall be charged at 5% on accounts not paid within 15 days of the invoice date"), some require that a certain state-of-affairs always obtains within a temporal interval (*e.g.*, "After opening a bank account, customers must keep a positive balance until bank charges are taken out."), some determine recurrent normative effects (*e.g.*, "each year from the 1st of June to the 1st of July immigrants entitled to come to Italy are obliged to pay taxes in the period they work in Italy."), and so forth.

Finally, as it is well known (see [17]), temporal coordinates apply to changes in the law, since the law is dynamic, and regulates its own dynamics: as time goes by, new legal norms are issued (according to competence norms), old norms are abrogated, the texts of some provisions are changed (so that the old norms expressed by the preexisting provision are terminated and new ones take their place). Usually changes have an impact only in the future, and effects generated by the preexisting norms before the change remain untouched However, some changes are retroactive, and have an impact on effects generated (or to be generated) before the change takes place.

The aim is this paper is provide a new fruitful synergy between argumentation formalism and Temporal Constraint Satisfaction Problem techniques able to model many of the temporal subtleties of the legal domain. After some initial attempts based in particular on logic programming (see [19]), more recently, legal dynamics has been modelled through extensions of Defeasible Logic [8, 9, 10, 11, 13]. This paper is novel on the following accounts.

Firstly, most previous work in the field is based on proof-theoretic approaches to temporal reasoning; on the contrary the present paper provides an argumentation-based framework. This is an advantage since, legal reasoning takes typically the form of arguments, so that an argument-based model allows for a more intuitive and transparent logical reconstruction of complex patterns of legal reasoning and

ICAIL '11 Pittsburgh, Pennsylvania USA

interaction; to the best of our knowledge, no attempt in the AI&Law community has been so far made in this direction.

Secondly, this paper proposes to integrate an argumentation framework with techniques from Temporal Constraint Satisfaction Problem (TCSP) (for a recent overview, see [5]). Accordingly, temporal information is represented as temporal constraints, where variables denote times points and constraints represent the temporal relations between time points. Temporal constraints allow us to exploit well-known, established and efficient techniques within an argumentation framework as well as to create a very expressive logical system able to handle complex queries regarding the temporal properties of argumentative conclusions.

Thirdly, while other papers have directly addressed controversial legal notions (such as the ideas of existence, force, efficacy, applicability of legal norms), the present contribution takes a more abstract view: we consider what norms are alive (able to produce legal effects) at what times, according to what temporal references. This applies both to substantive norms and to norms regulating their temporal dimension. The liveness of a legal norms can be made dependent of various legal predicates (existence, force, efficacy, applicability) which may be differently characterised in different legal systems and according to different doctrinal models.

The paper is organised as follows. In Section 2 we present our model for temporal aspects of legal norms. In Section 3 we introduce some preliminary formal definitions. Section 4 defines constrained arguments. Conflicts between arguments are dealt in Section 4, and their accommodation in Section 6. A fixed-point semantics is given in Section 7 and the associated dialogue game in Section 8. Some examples illustrate the expressiveness of the framework in Section 9. Related work is discussed in Section 10. Finally, we conclude and present some directions for future work.

2. LEGAL MODEL

It is a general tenet in the AI&Law literature that legal reasoning, being grounded in common sense reasoning, is defeasible and thus any formal account requires to deal with defeasibility. Accordingly, we frame normative theories as consisting of a set of defeasible rules and a set of priorities over rules (which establish their relative strength). In this perspective, rules naturally correspond to legal norms, while priorities represent the criteria used to solve legal conflicts. In our system, we assume that rules representing norms are expressions such as

$$\phi_0, \ldots, \phi_n \Rightarrow \psi$$

where $\phi_0, \ldots, \phi_n$ are the conditions under which the normative effect ψ follows. Legal norms are usually qualified by at least three temporal properties, such as the time when the norm is enacted, the time when the norm can produce legal effects and the time when normative effects hold [11].

For reasoning about temporal aspects in the law, that is basically to establish at what time a legal norm produces legal effects holding at what time, a number of different notions are used in legal doctrine, such as force, efficacy, applicability, etc. For our purposes, we prefer to manage temporal legal reasoning in a more abstract way: (i) we add a new temporal property, *liveness*, distinct from the temporal notions used in legal doctrine; (ii) we base abstract temporal reasoning upon this property, and make this property derivable from legal notions. Which and how many properties are required to derive liveness depends on the particular legal system: a legal system (or a legal theorist) may say that a norm is alive at T if it is valid and it is in force, it is effective, it is applicable and temporally relevant or scoped etc.

In our model, we view force and efficacy as mere preconditions for liveness. When a norm is in force and efficacious, it is by default alive. A norm in force may not be alive at a certain time, being declared inefficacious at that time, and an efficacious norm may not be alive at a certain time, being declared for not in force at that time.

A norm is alive during the intervals in which an instance of its antecedent causes an instance of its consequent. A norm may thus be alive in multiple intervals. For instance, a norm n may be alive from T_1 to T_2 and then suspended and alive again from T_3 to T_4. The liveness intervals of a norm n are determined by other norms (which must be live in their turn, to produce such impacts).

Let us assume a doctrinal model according to which (a) an abrogation deprives a norm of its force, (b) a suspension deprives a norm of its efficacy, and (c) liveness depends only on the combination of force and efficacy. Assume that the norm n becomes alive (being valid and effective, according to general rules on *vacatio legis*) at T_1, and that the temporal meta-norm n' is issued and becomes alive at T. The temporal norm n' can have different contents:

1. n': n is abrogated at time T, with $T_1 < T$;
2. n': n is abrogated at time T_2, with $T_1 < T_2 < T$;
3. n': n is suspended from T_2 to T_3, with $T_1 < T < T_2 < T_3$.

The first case is the normal one: n loses its liveness (as a consequence of losing its force) at the time T, which is posterior to the time T_1 when n became alive: from T on n' is no longer alive (and thus incapable of producing further effects).

The second case deals with retroactive abrogation: n loses its liveness at the time T_2 which is posterior to the time T_1 when n became alive, but anterior to the time when the abrogating norm n' became alive. Then, starting from time T (when abrogation takes place) we must assume that n' has not have been alive since T_2.

The third case deals with suspension: n is losing its liveness (as a consequence of losing its efficacy) during the interval between T_2 and T_3.

As this example shows, the liveness of a norm at a certain time T_1 is relative to the time T at which liveness is assessed. This happens since a norm's liveness intervals may change because of subsequent norms, which can also be retroactive: when a new norm n' retroactively modifies a condition for liveness of an antecedent norm n (such as the force or efficacy of n) then the result will be a change of the legal system in the past, a change that exists only from the time (from the viewpoint in the time) when the retroactive norm n' becomes live. Thus, in our model temporal viewpoints are considered in specifying when a norm is alive and when its antecedent is satisfied. This means that when a lawyer is assessing at T the regulation of a fact taking place at T_1 with $T_1 < T$, he will have to consider what norms where alive at T_1, according to the viewpoint of T. Indeed, the norms alive at T_1 from the viewpoint T may be different from the norm alive at T_1 from the viewpoint T_1.

For a norm n to produce its effect at a certain time point T_1, from the viewpoint of time T, it must be the case, that n is alive at T_1 from the viewpoint T and moreover, that that the norm's antecedent holds at T_1 from the viewpoint T. Accordingly, a simple norm r, with antecedent ϕ and consequent ψ, can be represented as follows:

$$\text{r} : \mathsf{Hold}^{T}\mathsf{Hold}^{T_1}\text{live}(\text{r}), \mathsf{Hold}^{T}\mathsf{Hold}^{T_1}\phi \Rightarrow \mathsf{Hold}^{T}\mathsf{Hold}^{T_1}\psi$$

Such a norm will produce its effect at a time T_1 from viewpoint T, only if r is alive (ready to fire) at T_1 from viewpoint T and the rule's antecedent is satisfied at the same time and viewpoint. Note that we have explicitly inserted the predicate live in the norms's body. An alternative solution would consist in developing a reasoning model that take liveness at the level of the inference rules, however this alternative would increase the complexity of the inference level while it would reduce the flexibility of the mathematical framework of our model (see next sections) in the sense that inference rules are assumed to be more stable than knowledge bases.

Whether a norm is alive depends on the conditions that determine liveness according to the legal system at issue, in particular force and efficacy. Therefore, in our model, we can still ask whether a norm is efficacious or in force at a certain time and from a certain viewpoint, so that all these dimensions remain, but they are ultimately used to establish whether the rule is alive.

A legal doctrine according to which liveness only depends on force and efficacy can be represented by the following general meta-norm:

$$\begin{aligned} \text{liveness} : \quad & \mathsf{Hold}^{T}\mathsf{Hold}^{T_1}\text{inforce}(n), \\ & \mathsf{Hold}^{T}\mathsf{Hold}^{T_2}\text{efficacious}(n), \\ & \Rightarrow \mathsf{Hold}^{T}\mathsf{Hold}^{T_2}\text{live}(n) \end{aligned}$$

Note that according to this meta-norm, liveness of a norm n starts with its efficacy. A norm which is no longer efficacious now may have been efficacious in the past. Consequently, from a certain perspective T we may possibly observe that a norm n, no longer alive at T (for instance, having been abrogated), was alive at an antecedent time T_2, since it was in force at time T_1 and efficacious at T_2. If n was retroactive, than T_2 (the efficacy time) may have been be anterior to to T_1 (the validity time). If n was postactive (according some doctrines norms in special cases can be considered efficacious also after the lose their force, having been abrogated) then T_2 may have been posterior to T_1.

In the next sections, we present a formal framework to account for the legal model sketched above.

3. PRELIMINARY DEFINITIONS

We first define our language including temporal literals, constraints and rules which connect temporal literals under constraints.

Definition 1. (Strong temporal literal) A strong temporal literal is a formula of the form $\mathsf{X}\phi$ where ϕ is a literal, and X is a (possibly empty) sequence of modalities Hold^{τ} such that τ is a (temporal) variable.

Intuitively, the formula $\mathsf{Hold}^{\tau}\psi$ expresses that ψ holds at time τ. The sequence of modalities allows us to account for temporal viewpoints. For example, $\mathsf{Hold}^{T_1}\mathsf{Hold}^{T_2}\psi$ expresses that from the viewpoint of time T_1, ψ holds at time T_2. In the remainder, the sequence X of a strong temporal literal $\mathsf{X}\phi$ is called its temporal sequence, and the temporal variables τ appearing in X are called the temporal perspectives of $\mathsf{X}\phi$.

Temporal knowledge is expressed through constraints associated to temporal literals to form constrained temporal literals.

Definition 2. Constraints, represented as γ, are any construct of the form $\tau \lhd \tau'$ where τ, τ' are first-order terms (that is, a variable, a constant or a function applied to terms) and $\lhd \in \{=, \neq, >, \geq, <, \leq\}$, or $\perp$.

We shall denote a possibly empty set of constraints as $\Gamma = \{\gamma_0, ..., \gamma_n\}$ and it stands as a conjunction of the constraints, that is, $\Gamma = \bigwedge_{i=0}^{i=n} \gamma_i$. We make use of numbers and linear arithmetic functions to build terms τ. Some sample constraints are $T < 8$ and $X < (Y + Z)$. To improve readability, constraints of the form $\{1 \leq X, X \leq 6\}$ will be written as $\{1 \leq X \leq 6\}$. The temporal constraint language can be restricted in order to keep computational complexity under control.

We make use of existing constraint satisfaction techniques [15, 16] to implement a $\mathsf{satisfy}$ predicate which checks if a given set of constraints admits one solution, that is, the predicate holds if the variables of the constraints admit at least one value which simultaneously fulfills all constraints:

Definition 3. Let $\Gamma = \{\gamma_0, \ldots, \gamma_n\}$ be a set of constraints. Predicate $\mathsf{satisfy}(\Gamma \cdot \sigma)$ holds if, and only if, $\gamma_0 \cdot \sigma \wedge \cdots \wedge \gamma_n \cdot \sigma$ holds for some substitution σ, or Γ is empty.[1]

We extend the use of the predicate to a set of constraints sets:

Definition 4. Let $\mathbf{\Gamma} = \{\Gamma_0, \ldots, \Gamma_n\}$ be a set of constraints sets, $\mathsf{satisfy}(\mathbf{\Gamma} \cdot \sigma)$ holds if, and only if, $\mathsf{satisfy}(\Gamma_i \cdot \sigma)$ holds for some Γ_i in $\mathbf{\Gamma}$, or $\mathbf{\Gamma}$ is empty.

For example, $\mathsf{satisfy}(\{\{T < 5, 9 < T\}, \{8 \leq T\}\} \cdot \sigma)$ holds for the substitution $\sigma = \{13/T\}$ (among others), $\mathsf{satisfy}(\{\{T < 5, 9 < T\}\} \cdot \sigma)$ does not hold for any substitution.

Constraints can be associated with strong literals, imposing restrictions on their variables.

Definition 5. (Constrained temporal literal) A constrained temporal literal is a formula of the form $\phi \circ \mathbf{\Gamma}$ such that ϕ is a strong temporal literal and $\mathbf{\Gamma}$ is a set of sets of constraints.

When there are no associated constraints then $\mathbf{\Gamma} = \{\{\}\}$ which is satisfiable.

A constrained temporal literal is the intensional form for all the ground instances whose temporal variables are substituted by any ground substitution satisfying the constraints. For example, if the domain of temporal variables are the natural numbers, then $\mathsf{Hold}^{T_1}\mathsf{Hold}^{T_2}\phi \circ \{\{6 < T_1 < 9, T_2 = 5\}\}$ stands for $\mathsf{Hold}^{7}\mathsf{Hold}^{5}\phi$ and $\mathsf{Hold}^{8}\mathsf{Hold}^{5}\phi$.

Definition 6. (Defeasible rule) A defeasible rule is an expression of the form:

$$r : \quad \phi_0, \ldots, \phi_j \Rightarrow \phi \circ \mathbf{\Gamma}$$

[1]A substitution σ is a finite and possibly empty set of pairs x/τ, where x is a variable and τ is a term. If $x/\tau \in \sigma$ then $x \cdot \sigma = \tau \cdot \sigma$ else $x \cdot \sigma = x$.

where r is a unique label to identify the rule, each antecedent ϕ_i is a strong temporal literal, the consequent ϕ is a strong literal and $\mathbf{\Gamma}$ is a set of satisfiable sets of constraints.

As for constrained temporal literals, if there are no constraints then $\mathbf{\Gamma} = \{\{\}\}$ which is assumed to be satisfiable. The variable appearing in a defeasible rule can take any value satisfying the constraints of the rule. Accordingly, a rule is the intensional form of all ground instances of the rules whose temporal variables are substituted by any ground substitution satisfying the constraints. For example, if the domain of temporal variables are the natural numbers, the rule

$$\text{r:} \quad \mathsf{Hold}^{T_1}\text{a} \Rightarrow \mathsf{Hold}^{T_2}\text{b} \circ \{\{T_1 \leq T_2\}\}$$

is an abbreviation for the following set of instantiated rules.

$$\left\{ \begin{array}{ll} \text{r:} & \mathsf{Hold}^{0}\text{a} \Rightarrow \mathsf{Hold}^{0}\text{b}, \\ \text{r:} & \mathsf{Hold}^{0}\text{a} \Rightarrow \mathsf{Hold}^{1}\text{b}, \\ \text{r:} & \mathsf{Hold}^{0}\text{a} \Rightarrow \mathsf{Hold}^{2}\text{b}, \\ \ldots & \\ \text{r:} & \mathsf{Hold}^{1}\text{a} \Rightarrow \mathsf{Hold}^{1}\text{b}, \\ \ldots & \end{array} \right\}$$

Conflicts between rules supporting contradictory conclusions are managed by a partial ordering $\succ$, capturing preference relations among rules saying when a rule may override the conclusion of another rule (see Section 5 on Conflicts).

A theory is a tuple $(\mathcal{R}, \mathcal{S}_{up})$ where $\mathcal{R}$ is a finite set of defeasible rules and $\mathcal{S}_{up}$ is a partial ordering over $\mathcal{R}$, such that all the strong temporal literals appearing in any rule in $\mathcal{R}$ have temporal sequences of equal length.

In the next section, we will see how to build arguments from a theory.

4. CONSTRAINED ARGUMENTS

Arguments, whose structure is a tree, are built from a theory by chaining defeasible rules. In the next definition, $\mathsf{conc}(A)$ is the top-conclusion of the argument A, $\mathsf{sub}(A)$ its sub-arguments, and $\mathsf{toprule}(A)$ its top-most rule.

Definition 7. (Argument) Arguments constructed from a theory $(\mathcal{R}, \mathcal{S}_{up})$ are recursively defined as follows:
1. $A_0, \ldots, A_n \Rightarrow \phi \circ \mathbf{\Gamma}$ if $A_0, \ldots, A_n$ are arguments such that

- $\mathsf{conc}(A) = \phi \circ \mathbf{\Gamma}$, $\mathsf{lit}(A) = \phi$, $\mathbf{\Gamma}(A) = \mathbf{\Gamma}$,
- there exists a defeasible rule $r : \phi_0, \ldots, \phi_n \Rightarrow \phi \circ \mathbf{\Gamma}_r$, with $\mathsf{unify}(\mathsf{lit}(A_i), \phi_i, \sigma)$ for $0 \leq i \leq n$,[2]
- $\mathbf{\Gamma} = (\Pi_0^n \mathbf{\Gamma}(A_i)) \times \mathbf{\Gamma}_r \cdot \sigma$.[3] If all the constraints in $\mathbf{\Gamma}$ are unsatisfiable, then $\mathbf{\Gamma}$ is replaced by $\{\{\bot\}\}$ else any unsatisfiable set of constraints in $\mathbf{\Gamma}$ is removed.
- $\mathsf{sub}(A) = \{A\} \cup \mathsf{sub}(A_0) \cup \ldots \cup \mathsf{sub}(A_n)$,
- $\mathsf{toprule}(A) = r$.

2. $A_0, \ldots, A_n \Rightarrow \phi \circ \mathbf{\Gamma}$ if $A_0, \ldots, A_n$ are arguments such that

- $\mathsf{conc}(A) = \phi \circ \mathbf{\Gamma}$, $\mathsf{lit}(A) = \phi$, $\mathbf{\Gamma}(A) = \mathbf{\Gamma}$,
- $\mathsf{conc}(A_i) = \phi \circ \mathbf{\Gamma}_i$,
- $\mathbf{\Gamma} = \cup_0^n \mathbf{\Gamma}(A_i)$,
- the set $\{A_0, \ldots, A_n\}$ is maximal w.r.t. set inclusion,
- $\mathsf{sub}(A) = \{A\} \cup \mathsf{sub}(A_0) \cup \ldots \cup \mathsf{sub}(A_n)$,
- $\mathsf{toprule}(A)$ is undefined.

The first item expresses modus ponens while the second item accounts for situations where different arguments supporting a conclusion at different instants can be merged into one argument supporting the conclusion at the union of these instants (on the issue of accruals of arguments, see [20]). See Section 9 for some illustrations of arguments.

Notice that the above definition of arguments does not require the satisfiability of constraints. We say that an argument is valid if, and only if, its constraints are satisfiable.

Definition 8. (Valid argument) An argument A is valid if, and only if, $\mathbf{\Gamma}(A)$ is satisfiable.

We will see in Section 6 that an argument can be invalided when it is attacked by contradictory arguments. Conflicts between valid arguments supporting contradictory conclusions are defined in the next section.

5. CONFLICTS BETWEEN ARGUMENTS

In most common argumentation systems, the rebuttal and undercutting of one argument by another are considered. A rebutting argument provides a reason for contradicting a conclusion of another argument, while an undercutting argument undermines the support of a conclusion without contradicting it. For our purposes, we deal with rebuttals only.

First, contradictory temporal literals have to be defined. Let $\mathcal{L}$ be a set of literals, ψ an atom, and X, Y (possibly empty) temporal sequences of the same length. We assume the availability of a function $-$, which works with $\mathcal{L}$, such that $-\mathsf{X}\psi = \mathsf{Y}\neg\psi$ and $-\mathsf{X}\neg\psi = \mathsf{Y}\psi$. For example, $-\mathsf{Hold}^W\mathsf{Hold}^X\psi = \mathsf{Hold}^Y\mathsf{Hold}^Z\neg\psi$.

Conflicts between contradictory argument conclusions are resolved on the basis of preferences over arguments. We settle preferences over arguments using a simple last-link ordering over arguments according which an argument A is preferred over another argument B, denoted as $A \succ B$, if, and only if, the rule $\mathsf{toprule}(A)$ is preferred to the rule $\mathsf{toprule}(B)$ (i.e. $\mathsf{toprule}(A) \succ \mathsf{toprule}(B)$). While this simple last-link ordering is sufficient for our purposes, other preferences over arguments could be certainly settled.

Definition 9. (Successful rebuttal) An argument B successfully rebuts an argument A on a subargument A' if, and only if,

- $\exists A' \in \mathsf{sub}(A)$, such that $\mathsf{unify}(\mathsf{conc}(B), -\mathsf{conc}(A'), \sigma_1)$,
- $\mathsf{toprule}(A')$ is defined and $A' \not\succ B$,
- $\mathsf{satisfy}(\mathbf{\Gamma}(A') \times \mathbf{\Gamma}(B) \cdot \sigma_1 \cdot \sigma)$ holds.

The last condition of the above definition specifies that a rebuttal is possible only if a temporal overlap of the contradictory conclusions exists w.r.t. the associated constraints.

[2]A substitution σ is an unifier of two terms τ_1 and τ_2 if $\tau_1 \cdot \sigma = \tau_2 \cdot \sigma$. We have that $\mathsf{unify}(\tau_1, \tau_2.\sigma)$ holds if and only if $\tau_1 \cdot \sigma = \tau_2 \cdot \sigma$ for some σ.

[3]The product of two sets of sets $\mathbf{\Gamma} = \{\Gamma_1, \ldots, \Gamma_n\}$ and $\mathbf{\Gamma}' = \{\Gamma'_1, \ldots, \Gamma'_m\}$ is denoted as $\mathbf{\Gamma} \times \mathbf{\Gamma}' = \{\Gamma_i \cup \Gamma'_j | i \leq n, j \leq m\}$. The product is commutative, $\mathbf{\Gamma} \times \mathbf{\Gamma}' = \mathbf{\Gamma}' \times \mathbf{\Gamma}$.

For example, no rebuttal exists between the arguments supporting the conclusions $\mathsf{Hold}^{X_1}\neg\mathsf{Hold}^{X_2}\psi \circ \{\{X_2 < 5\}\}$ and $\mathsf{Hold}^{Y_1}\mathsf{Hold}^{Y_2}\psi \circ \{\{5 < Y_2\}\}$ because $\mathsf{satisfy}(\{\{X_2 < 5, 5 < Y_2\}\} \cdot \sigma_1 \cdot \sigma)$ does not hold (with the substitution σ_1 such that $\mathsf{unify}(\mathsf{Hold}^{X_1}\mathsf{Hold}^{X_2}\psi, \mathsf{Hold}^{Y_1}\mathsf{Hold}^{Y_2}\psi, \sigma_1))$. This condition also implies that a rebuttal can be settled only if the rebutted and the rebutting arguments are both valid. Indeed, if $\mathbf{\Gamma}(A')$ or $\mathbf{\Gamma}(B)$ are unsatisfiable, then $\mathbf{\Gamma}(A') \times \mathbf{\Gamma}(B)$ is unsatisfiable.

In the remainder, and in order to comply with conventional terminology, we say that an argument B defeats an argument A on a subargument A' if, and only if, B successfully rebuts A on A'.

In the next section, the concept of defeated argument is completed by the notion of *accommodated* and *crushed* argument.

6. ARGUMENT ACCOMMODATION

The accommodation is achieved via the addition of constraints to a defeated argument so that it is no longer rebutted. More precisely, if an argument B defeats an argument A, then it is required that the accommodated form of A by B, denoted $A \lhd B$, is such that the set of constraints $\mathbf{\Gamma}(A \lhd B) \times \mathbf{\Gamma}(B) \cdot \sigma_1$ becomes unsatisfiable with the substitution σ_1 such that $\mathsf{unify}(\mathsf{conc}(A \lhd B), -\mathsf{conc}(B), \sigma_1)$.

The constraints $\mathbf{\Gamma}$ to be added to $\mathbf{\Gamma}(A)$ to obtain $\mathbf{\Gamma}(A \lhd B) = \mathbf{\Gamma}(A) \times \mathbf{\Gamma}$, is such that the solution set of $\mathbf{\Gamma}$ is the complement solution set of $\mathbf{\Gamma}(B)$. The accommodation concerns only the solution set of the temporal perspective variables of a temporal literal. Consequently, and more precisely, the solution set of the constraints $\mathbf{\Gamma}$ is the complement of the $\mathbf{\Gamma}(B)$'s solution set for the perspectives variables of the defeated temporal literal. In other words, the $\mathbf{\Gamma}$'s solution set is the complement of the $\mathbf{\Gamma}(B)$'s solution set projected onto the temporal perspectives, or equivalently, the complement of the $\mathbf{\Gamma}(B)$'s solution set in which non-perspective variables are eliminated.

Let $\mathbf{\Gamma}$ be a set of constraint sets, $\mathrm{var}(\mathbf{\Gamma})$ the set of variables appearing in $\mathbf{\Gamma}$, and $\mathcal{T}$ a set of variables, we denote $\mathbf{\Gamma}^{\mathcal{T}}$ the set $\mathbf{\Gamma}$ in which the variables $\mathrm{var}(\mathbf{\Gamma})\backslash\mathcal{T}$ are eliminated from any set in $\mathbf{\Gamma}$ (See [15] for algorithms on variable elimination). We assume that for any $\mathcal{T}$, $\{\{\bot\}\}^{\mathcal{T}} = \{\{\bot\}\}$. For instance, $\{\{0 \leq X \leq Y\}\}^{Y} = \{\{0 \leq Y\}\}$.

The complement of Γ's solution is obtained via the negation of constraints. Let γ be a constraint, its negated form $\neg\gamma$ is defined as:

- $\neg(\tau = \tau') \equiv (\tau \neq \tau')$, and $\neg(\tau \neq \tau') \equiv (\tau = \tau')$,
- $\neg(\tau > \tau') \equiv (\tau \leq \tau')$, and $\neg(\tau \leq \tau') \equiv (\tau > \tau')$,
- $\neg(\tau < \tau') \equiv (\tau' \leq \tau)$, and $\neg(\tau \leq \tau') \equiv (\tau' < \tau)$.

Since a non-empty set of constraints $\Gamma = \{\gamma_0, \ldots, \gamma_n\}$ stands as a conjunction of the constraints, that is, $\Gamma = \bigwedge_0^n \gamma_i$, its negated form $\neg\Gamma$ is the disjunction of the negated constraints, that is, $\neg\Gamma = \bigvee_0^n \neg\gamma_i$. Using set notation, we have $\neg\Gamma = \bigcup_0^n \{\{\neg\gamma_i\}\}$. We assume that $\{\neg\bot\} = \{\}$, and $\neg\{\} = \{\bot\}$. Let $\mathbf{\Gamma} = \{\Gamma_0, \ldots, \Gamma_n\}$ be a set of constraint sets, $\neg\mathbf{\Gamma}$ is $\Pi_0^n \neg\Gamma_i$. Elimination of constraints and complement constraints sets are used in the next definition of accommodation w.r.t. rebuttal.

Definition 10. (Accommodation w.r.t. rebuttal) Let A and B two arguments such that B successfully rebuts A on A, the accommodated form of A w.r.t. B, denoted $A \lhd B$, is $A \cdot \{\mathbf{\Gamma}(A \lhd B)/\mathbf{\Gamma}(A)\}$[4] such that

$$\mathbf{\Gamma}(A \lhd B) = \mathbf{\Gamma}(A) \times \neg(\Gamma^{\mathcal{T}}(B)) \cdot \sigma$$

where $\mathcal{T}$ are the temporal perspectives of $\mathsf{conc}(B)$ and σ is a substitution such that $\mathsf{unify}(\mathsf{conc}(B), -\mathsf{conc}(A), \sigma)$. If all the sets in $\mathbf{\Gamma}(A \lhd B)$ are unsatisfiable then $\mathbf{\Gamma}(A \lhd B)$ is replaced by $\{\{\bot\}\}$, else any unsatisfiable set in $\mathbf{\Gamma}(A \lhd B)$ is removed.

Notice that an accommodation may occur only between valid arguments since the rebutting and rebutted arguments are necessarily valid (see definition on successful rebuttals).

If an argument B defeats an argument A on A', then A' in A is substituted by $A' \lhd B$ in A, and the accommodated constraints $\mathbf{\Gamma}(A' \lhd B)$ have to be propagated to the top-most conclusion $\mathsf{conc}(A)$. In other words, if A is accommodated by B on a sub-argument A' then the accommodated form of A w.r.t. B is $A \cdot \{A'/A' \lhd B\}$ (the argument A where the sub-argument A' is substituted by $A' \lhd B$) such that the third item of Definition 7 on argument is respected.

If an argument A is accommodated by B, then $A \lhd B$ does not conflict with B since $\mathbf{\Gamma}(A \lhd B) \times \mathbf{\Gamma}(B)$ is unsatisfiable, and two exclusive cases exist, either (i) $\mathbf{\Gamma}(A \lhd B)$ is satisfiable thus $A \lhd B$ is valid, we say that $A \lhd B$ is *not completely* defeated, or (ii) $\mathbf{\Gamma}(A \lhd B)$ is unsatisfiable thus $A \lhd B$ is invalid, we say that A is *crushed* by B.

Definition 11. (Crushed argument) An argument A is crushed by an argument B if, and only if, $A \lhd B$ is invalid.

Remark that even if $A' \lhd B$ is not crushed, $A \lhd B$ can turn out to be crushed (i.e. $\mathbf{\Gamma}(A \lhd B)$ is not satisfiable) after the propagation of the accommodated constraints to the top-most conclusion of A.

In the next section, we will study the interactions amongst several arguments, and give a fixed-point semantics.

7. FIXED-POINT SEMANTICS

As a first investigation, the proposed semantics is based on the Dung's grounded semantics [6] though other semantics could be certainly investigated. Accordingly, we define an argumentation framework as follows.

Definition 12. (Argumentation framework) An argumentation framework is a pair $AF = \langle \mathcal{A}, \mathsf{defeat} \rangle$ where $\mathcal{A}$ is a set of valid arguments, and $\mathsf{defeat} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation of defeat. An argument B defeats an argument A if, and only if, $(A, B) \in \mathsf{defeat}$.

A set $\mathcal{S}$ of arguments is said to be conflict-free if there is no arguments A, B in $\mathcal{S}$ such that B defeats A.

In Dung's approach, an argument A is acceptable w.r.t. a set $\mathcal{S}$ of arguments if, and only if, for any argument B defeating A, B is defeated by $\mathcal{S}$. This definition has to be slightly adapted in order to take into account the accommodation of arguments. First, we define the accommodation of an argument A w.r.t. a set $\mathcal{B}$ of defeating arguments.

[4] $A \cdot \{\mathbf{\Gamma}'/\mathbf{\Gamma}(A)\}$ denotes the argument A where the constraints $\mathbf{\Gamma}(A)$ are substituted by $\mathbf{\Gamma}'$.

Definition 13. Let A be an argument, and $\mathcal{B}$ a set of arguments. The accommodated form of A w.r.t. $\mathcal{B}$, denoted $A \lhd \mathcal{B}$, is A_∞ recursively defined as:

- $A_0 = A$,
- $A_{i+1} = A_i \lhd B$ such that $B \in \mathcal{B}$ defeats A_i.

According to this definition, if no arguments in $\mathcal{B}$ defeats A, then $A \lhd \mathcal{B} = A$ because we have $A = A \lhd \emptyset$.

Similarly to the case in which an argument A is accommodated by a single argument B, two exclusive cases exist, either (i) $A \lhd \mathcal{B}$ is valid, we say $A \lhd \mathcal{B}$ is *not completely* defeated, or (ii) $A \lhd \mathcal{B}$ is invalid, we say that A is *crushed* by $\mathcal{B}$ (A is crushed by $\mathcal{B}$ if, and only if, $A \lhd \mathcal{B}$ is invalid).

However, remark that if $A' \lhd \mathcal{B}$ is not crushed (A' is a sub-argument of A), $A \lhd \mathcal{B}$ can turn out to be crushed as the propagation of the accommodated constraints to the topmost conclusion can imply that $\mathbf{\Gamma}(A \lhd \mathcal{B})$ turns out to be unsatisfiable. In any case, if a sub-argument A' of A is crushed by $\mathcal{B}$, i.e. $A' \lhd \mathcal{B}$ is crushed, then $A \lhd \mathcal{B}$ is crushed.

Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of arguments, the accommodated form of $\mathcal{A}$ w.r.t. $\mathcal{B}$, denoted $\mathcal{A} \lhd \mathcal{B}$ is $\{A \lhd \mathcal{B} | \forall A \in \mathcal{A}\}$. In the remainder, and as for notation, we assume that $A \lhd \mathcal{B} \lhd \mathcal{S}$ stands for $A \lhd (\mathcal{B} \lhd \mathcal{S})$.

Definition 14. (Acceptable argument) Let A be an argument, $\mathcal{S}$ a set of arguments, and a set $\mathcal{B} = \{B |\ B$ defeats $A\}$. The accommodated argument $A \lhd \mathcal{B} \lhd \mathcal{S}$ is acceptable w.r.t. $\mathcal{S}$ if, and only if, $A \lhd \mathcal{B} \lhd \mathcal{S}$ is valid.

This definition of acceptable arguments embeds Dung's definition of acceptable grounded instance arguments $A \lhd \mathcal{B} \lhd \mathcal{S} \cdot \sigma$ where the substitution σ is grounded.[5] To see it, notice first that if an argument $B \cdot \sigma$ defeats an argument $A \cdot \sigma$ where σ is grounded, then $A \lhd B \cdot \sigma$ is crushed. Consider the valid grounded argument $A \lhd \mathcal{B} \lhd \mathcal{S} \cdot \sigma$, we have two cases: (i) either it exists no valid argument $B \cdot \sigma$ in $\mathcal{B} \cdot \sigma$, that is, $\mathcal{B} \cdot \sigma = \emptyset$, thus $A \lhd \mathcal{B} \lhd \mathcal{S} \cdot \sigma = A \lhd \emptyset \lhd \mathcal{S} \cdot \sigma = A \cdot \sigma$ which is acceptable w.r.t. $\mathcal{S} \cdot \sigma$; (ii) or $A \lhd \mathcal{B} \lhd \mathcal{S} \cdot \sigma$ is crushed by an argument $B \cdot \sigma$ in $\mathcal{B} \cdot \sigma$ crushed by an argument $S \cdot \sigma$ in $\mathcal{S} \cdot \sigma$, thus $A \lhd \mathcal{B} \lhd \mathcal{S} \cdot \sigma$ is accepted w.r.t. $\mathcal{S} \cdot \sigma$.

The notion of a set of arguments defending an argument is captured by the definition of characteristic function which is slightly adapted for our purposes w.r.t. Dung's definition (see [6]). We denote by $\mathcal{A}^\lhd$ the set of non-accommodated arguments and accommodated arguments of a set of arguments $\mathcal{A}$.

Definition 15. (Characteristic function) The characteristic function, denoted F_{AF}, of an argumentation framework $AF = (\mathcal{A}, \mathsf{defeat})$, is defined as follows:

- $F_{AF} : 2^{\mathcal{A}^\lhd} \rightarrow 2^{\mathcal{A}^\lhd}$,
- $F_{AF}(\mathcal{S}) = \{A |\ A$ is acceptable w.r.t. $\mathcal{S} \subseteq \mathcal{A}^\lhd\}$.

In other words, $F(\mathcal{S})$ represents all the arguments defended by the set of (possibly accommodated) arguments $\mathcal{S}$. We recall that the fixed-point of the characteristic function F_{AF} of an argumentation framework $AF = (\mathcal{A}, \mathsf{defeat})$ is $\bigcup_{i=1}^{\infty}(F^i)$ where:

- $F^1_{AF} = F_{AF}(\emptyset)$,
- $F^{i+1}_{AF} = F^i_{AF} \cup F_{AF}(F^i_{AF})$.

The definition of the grounded extension w.r.t. the fixed-point of the characteristic function remains the same as in [6]:

Definition 16. (Grounded extension) The grounded extension of an argumentation framework AF is the least fixed-point of F_{AF}.

We say that an accommodated argument is justified if it belongs to the grounded extension.

Theorem 1. If an accommodated argument is justified then it is valid.

PROOF. If an accommodated argument $A^\lhd$ is justified then $A^\lhd$ is acceptable w.r.t. a set of argument. If $A^\lhd$ is acceptable w.r.t. a set of argument then it is valid by definition. □

In the next section, we propose a dialogue game to check whether an argument or an accommodated version belongs to the grounded extension.

8. DIALOGUE GAME

The dialogue game we propose is inspired by the dialogue game given in [21] used in grounded argumentation frameworks to compute whether a grounded instance argument is justified, i.e. belongs to the grounded extension. In [21], a dialogue between a proponent pro and a opponent opp is a finite nonempty sequence of moves move$(i) = (Player_i, A_i)$ with $0 < i$ such that

- $Player_i =$ pro if i is odd, and $Player_i =$ opp if i is even,
- the grounded instance argument A_i defeats the grounded instance argument A_{i-1},
- if $Player_i = Player_j =$ pro then A_i is different than A_j.

A dialogue tree is a finite tree of moves such that each branch is a dialogue and if $Player_i =$ pro then the children of $move_i$ are all the defeaters of A_i. A player wins a dialogue tree if the other player cannot move, and a player wins a dialogue tree if it wins all the branches of the dialogue tree. A ground instance argument A is provably justified if, and only if, there is a dialogue tree with A as its root, and won by the proponent. It is proven that all provably justified ground instance argument are justified, i.e. belongs to the grounded extension.

For our purposes, we present a dialogue game slightly modified to take into account accommodations of arguments.

Definition 17. (Dialogue tree) A dialogue tree is a finite tree where each node is a move move(i, j) = move$(Player_i, A_j, A_j^\lhd)$, $(0<i)$ with n children move$(i + 1, k)$ = move$(Player_{i+1}, B_k, B_k^\lhd)$ (0≤n) such that:

- $Player_i =$ pro if i is odd, and $Player_i =$ opp if i is even,
- $\{B_1, \ldots, B_n\}$ are all defeaters of A_j,

[5] As for notation, we assume that $A \lhd \mathcal{B} \lhd \mathcal{S} \cdot \sigma = A \cdot \sigma \lhd \mathcal{B} \cdot \sigma \lhd \mathcal{S} \cdot \sigma$, and $\mathcal{B} \cdot \sigma = \{B \cdot \sigma | B \in \mathcal{B}, \mathbf{\Gamma}(B \cdot \sigma)$ is satisfiable $\}$.

- $A_j^{\triangleleft} = A_j \triangleleft \{B_1^{\triangleleft}, \ldots, B_n^{\triangleleft}\}$,
- in a branch, if $Player_p = Player_q =$ pro then A_p is different than A_q.

In the above definition, the second item implies that all non-accommodated arguments are valid. If an argument A has no defeater, then its associated move is move$(Player_i, A, A)$, that is, the argument A is not accommodated since it is not rebutted by any argument. See Section 9 for simple examples of dialogue tree.

Definition 18. The proponent *rides out* a dialogue tree if, and only if, the accommodated root argument $A^{\triangleleft}$ is not crushed. The proponent *wins* a dialogue tree if, and only if, the accommodated root argument $A^{\triangleleft}$ unifies with the root argument A.

An accommodated argument $A^{\triangleleft}$ is provably justified if, and only if, there is a dialogue tree with $A^{\triangleleft}$ as its accommodated root argument, and the proponent rides out the dialogue.

Theorem 2. An accommodated argument $A^{\triangleleft}$ is provably justified if, and only if, the accommodated root argument $A^{\triangleleft}$ is not crushed.

Theorem 3. An accommodated argument $A^{\triangleleft}$ is provably justified if, and only if, the accommodated root argument $A^{\triangleleft}$ is valid.

The proofs are trivially based on the definition of a player riding out a dialogue tree, and the definition of crushed arguments.

Theorem 4. Any provably justified argument is justified, i.e. belongs to the grounded extension.

PROOF. We prove by induction that any valid accommodated argument $A_i^{\triangleleft}$ in the dialogue tree is justified by showing that $A_i^{\triangleleft}$ belongs to the fixed-point of the characteristic function.

As induction basis, we note that a leaf argument A of a dialogue tree (which is trivially provably justified) has no defeating argument if, and only if, A is acceptable w.r.t. the empty set, and thus $A \in F_{AF}^1 = F_{AF}(\emptyset)$, therefore, A belongs to the grounded extension, A is justified.

Let A_{i+1} be an argument defeated by its children $\{A_i^1, \ldots, A_i^n\}$. As induction hypothesis, suppose that the set $\mathsf{J}(A_i^{\triangleleft})$ of valid arguments in $\{A_i^{1\triangleleft}, \ldots, A_i^{n\triangleleft}\}$ is provably justified, and belongs to the grounded extension (thus, $\mathsf{J}(A_i^{\triangleleft}) \subseteq F_{AF}^i$). By definition of the dialogue tree, $A_{i+1}^{\triangleleft} = A_{i+1} \triangleleft \mathsf{J}(A_i^{\triangleleft})$. However, by the induction hypothesis, we have $A_{i+1} \triangleleft \mathsf{J}(A_i^{\triangleleft}) \in F_{AF}^{i+1}$. Consequently, $A_{i+1}^{\triangleleft}$ is justified if, and only if, $A_{i+1}^{\triangleleft}$ is valid, that is, $A_{i+1}^{\triangleleft}$ is provably justified. □

9. LEGAL EXAMPLES

In this section we illustrate how the system works by two simple scenarios that typically occur in the law. The first scenario illustrates a simple deadline scenario while the second shows the formalisation of a retroactive legal change. In the remainder, some temporal variables are renamed and some constraint sets are simplified for the sake of readability when arguments and accommodations are computed.

9.1 Deadline

In this section, we illustrate how the system can model normative deadlines. We follow here the conceptual analysis developed in [8]. The idea of normative deadline usually refers to temporally-qualified obligations. Consider the following example.

> The payment terms for customers shall be in full upon receipt of invoice. Interest shall be charged at 5% on accounts not paid within 15 days after the receipt date of the invoice.

This example states an obligation for customers to pay within 15 days upon the receipt of an invoice. To keep our presentation light, we do not need to explicitly introduce operators to capture obligations ([8, 13]). Accordingly, an expression like Hold^{T_1}oblpay, meaning that it is obligatory to pay at time T_1, is logically treated here as a standard strong temporal literal. The policy given in the example is represented by the following rules and superiority relations over rules.

a: $\mathsf{Hold}^{T_1}\text{invoice} \Rightarrow \mathsf{Hold}^{T_2}\text{oblpay} \circ \{\{T_1 < T_2\}\}$
b: $\mathsf{Hold}^{T_1}\text{invoice}, \mathsf{Hold}^{T_2}\text{pay} \Rightarrow \mathsf{Hold}^{T_3}\neg\text{oblpay}$
$\circ\{\{T_1 < T_2 < T_3\}\}$
c: $\mathsf{Hold}^{T_1}\text{invoice}, \mathsf{Hold}^{T_2}\text{oblpay} \Rightarrow \mathsf{Hold}^{T_3}\text{oblpay5\%}$
$\circ\{\{T_1 + 15 \leq T_2 < T_3\}\}$
d: $\mathsf{Hold}^{T_1}\text{invoice}, \mathsf{Hold}^{T_2}\text{pay5\%} \Rightarrow \mathsf{Hold}^{T_3}\neg\text{oblpay5\%}$
$\circ\{\{T_1 + 15 < T_2 < T_3\}\}$

$$\text{b} \succ \text{a}, \text{d} \succ \text{c}$$

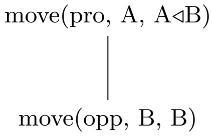
The first rule (a) states that, after the invoice is received, the customer is obliged to pay. The obligation persists indefinitely and does not terminate unless it is complied with (rule b). Of course, the obligation to pay persists beyond the deadline if it is not fulfilled, but after the deadline the non-compliance triggers another obligation (rule c), stating to pay 5% of the due amount. This obligation terminates if the 5% are paid (rule d). Notice that, for the sake of readability, rules do not have any predicate expressing liveness as the flexibility of our framework allows us to do it. Suppose that the premises are formalised by the following set of rules with empty antecedents:

p1: $\Rightarrow \mathsf{Hold}^{X}\text{invoice} \circ \{\{X = 1\}\}$
p2: $\Rightarrow \mathsf{Hold}^{Y}\text{pay} \circ \{\{Y = 16\}\}$
p3: $\Rightarrow \mathsf{Hold}^{Z}\text{pay5\%} \circ \{\{Z = 20\}\}$

We can build the following valid arguments from the above rules.

P1 :		$\Rightarrow \mathsf{Hold}^{X}\text{invoice} \circ \{\{X = 1\}\}$
P2 :		$\Rightarrow \mathsf{Hold}^{Y}\text{pay} \circ \{\{Y = 16\}\}$
P3 :		$\Rightarrow \mathsf{Hold}^{Z}\text{pay5\%} \circ \{\{Z = 20\}\}$
A :	P1	$\Rightarrow \mathsf{Hold}^{T_2}\text{oblpay} \circ \{\{X = 1, X < T_2\}\}$
B :	P1, P2	$\Rightarrow \mathsf{Hold}^{T_3}\neg\text{oblpay}$ $\circ\{\{X = 1, Y = 16, X < Y < T_3\}\}$
C :	P1, A	$\Rightarrow \mathsf{Hold}^{T_3}\text{oblpay5\%}$ $\circ\{\{X = 1, X + 15 \leq T_2 < T_3\}\}$
D :	P1, P3	$\Rightarrow \mathsf{Hold}^{T_3}\neg\text{oblpay5\%}$ $\circ\{\{X = 1, Z = 20, X + 15 < Z < T_3\}\}$

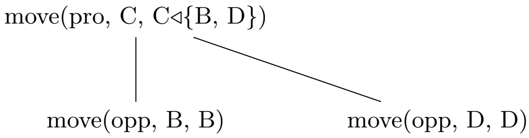
A simple dialogue game can be drawn:



where A◁B : P1 $\Rightarrow \mathsf{Hold}^{T_2}$oblpay $\circ \{\{1 < T_2 \leq 16\}\}$. The constraints $\mathbf{\Gamma}$(A◁B) are computed as follows:

$$\begin{aligned}&\mathbf{\Gamma}(A) \times \neg\mathbf{\Gamma}^{\{T_2\}}(B)\\ &= \{\{X = 1, X < T_2\}\} \times \neg\{\{16 < T_2\}\}\\ &= \{\{1 < T_2\}\} \times \{\{T_2 \leq 16\}\}\\ &= \{\{1 < T_2 \leq 16\}\}\end{aligned}$$

In words, the computation of the argument A◁B concludes, as expected, that the obligation to pay holds between 1 and 16 because the obligation terminates after its compliance at 16. Another dialogue can be drawn:



By definition 13, the argument C◁{B, D} is decomposed as (C◁B)◁D.

$$\text{C◁B} : \quad \text{P1, A◁B} \;\Rightarrow \mathsf{Hold}^{T_3}\text{oblpay5\%} \circ \{\{1 < T_2 \leq 16, 16 \leq T_2 < T_3\}\}$$

The constraints of the accommodated argument $C \lhd \{B, D\}$ are computed as follows:

$$\begin{aligned}&\mathbf{\Gamma}(C \lhd \{B, D\})\\ &= \mathbf{\Gamma}(C \lhd B) \times \neg\mathbf{\Gamma}^{\{T_3\}}(D)\\ &= \{\{1 < T_2 \leq 16, 16 \leq T_2 < T_3\}\} \times \neg\{\{\{20 < T_3\}\}\\ &= \{\{1 < T_2 \leq 16, 16 \leq T_2 < T_3\}\} \times \{\{T_3 \leq 20\}\}\\ &= \{\{1 < T_2 \leq 16, 16 \leq T_2 < T_3 \leq 20\}\}\end{aligned}$$

Thus we obtain the accommodated argument C◁{B, D} which is valid:

$$\text{C◁\{B,D\}} : \quad \text{P1, A◁B} \Rightarrow \mathsf{Hold}^{T_3}\text{oblpay5\%} \circ \{\{1 < T_2 \leq 16, 16 \leq T_2 < T_3 \leq 20\}\}$$

The computation of the dialogue shows that, as expected, the obligation to pay 5% of the due amount starts at time 16 and ends at 20 when the interests are paid. The argument C◁{B, D} is valid, thus the proponent rides out the dialogue.

9.2 Retroactive Abrogation

In this section, we illustrate how the framework can model a retroactive abrogation. Consider a norm n according which the occurence of a certain event triggers an obligation for a certain action. Let's represent this norm by the following rule (we use double temporal indexes to model viewpoints):

$$\text{n:} \quad \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{live(n)}, \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{event} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\text{obligation} \circ \{\{T_1 < T_2\}\}$$

Let us also assume the general liveness rule:

$$\text{liveness:} \quad \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{inforce(N)}, \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\text{efficacious(N)} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\text{live(N)} \circ \{\{\}\}$$

Let us assume that a norm comes into force and becomes efficacious when it is enacted, and persists being in force and efficacious by default after that (in fact in many legal systems a time period is required, called vacatio legis, between enactment and beginning force/efficacy, but for simplicity's sake we leave that out).

$$\text{f:} \quad \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{enacted(N)} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\text{inforce(N)} \circ \{\{T_1 \leq T_0, T_1 \leq T_2\}\}$$

$$\text{e:} \quad \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{enacted(N)} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\text{efficacious(N)} \circ \{\{\{T_1 \leq T_0, T_1 \leq T_2\}\}$$

Assume that the following facts hold, stating that from the perspective of T_0, the norm n was enacted at time 1 and n's antecedent event happened at time 9:

$$\text{p1:} \quad \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{enacted(n)} \circ \{\{T_1 = 1\}\}$$

$$\text{p2:} \quad \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{event} \circ \{\{T_1 = 9\}\}$$

Then we have arguments that after time 1, norm n is in force and efficacious, and therefore alive.

$$\begin{aligned}&\text{P1:} & & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{enacted(n)} \circ \{\{T_1 = 1\}\}\\ &\text{F:} & \text{P1} & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{F_1}\text{inforce(n)} \circ \{\{1 \leq T_0, 1 \leq F_1\}\}\\ &\text{E:} & \text{P1} & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{E_1}\text{efficacious(n)} \circ \{\{1 \leq T_0, 1 \leq E_1\}\}\\ &\text{L:} & \text{F,E} & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{E_1}\text{live(n)} \circ \{\{1 \leq T_0, 1 \leq F_1, 1 \leq E_1\}\}\end{aligned}$$

By applying the rule n to the event we can infer the obligation.

$$\begin{aligned}&\text{P2:} & & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{E_1}\text{event} \circ \{\{E_1 = 9\}\}\\ &\text{N:} & \text{L, P2} & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\text{obligation} \circ \{\{1 \leq T_0, 1 \leq F_1, 1 \leq E_1, E1 = 9, T_1 < T_2\}\}\end{aligned}$$

Consider a norm n' issued at time 10, which abrogates retroactively the norm n from time 8:

$$\text{n':} \quad \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{live(n')} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\neg\text{efficacious(n)} \circ \{\{8 \leq T_2\}\}$$

$$\text{p1':} \quad \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{enacted(n')} \circ \{\{T_1 = 10\}\}$$

with n'≻e. We assume that abrogation terminates efficacy. The same result would be obtained by assuming that abrogation terminates force, but efficacy is implicitly terminated when force terminates. Then we have the following arguments, corresponding to the arguments above concerning n, and concluding that at time 10 rule n is in force and efficacious, and therefore alive.

$$\begin{aligned}&\text{P1':} & & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\text{enacted(n')} \circ \{\{T_1 = 10\}\}\\ &\text{F':} & \text{P1'} & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{F_1}\text{inforce(n')} \circ \{\{10 \leq T_0, 10 \leq F_1\}\}\\ &\text{E':} & \text{P1'} & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{E_1}\text{efficacious(n')} \circ \{\{10 \leq T_0, 10 \leq E_1\}\}\\ &\text{L':} & \text{F', E'} & \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{E_1}\text{live(n')} \circ \{\{10 \leq T_0, 10 \leq F_1, 10 \leq E_1\}\}\end{aligned}$$

By applying rule n' we can conclude that the norm n is not efficacious since time 8.

$$\text{N':} \quad \text{L'} \;\Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_1}\neg\text{efficacious(n)} \circ \{\{10 \leq T_0, 10 \leq F_1, 10 \leq E_1, 8 \leq T_1\}\}$$

The argument E is rebutted by N', thus the following accommodations are possible:

$$\begin{array}{ll} \text{E}\lhd\text{N'}: & \text{P1} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{E_1}\text{live(n)} \\ & \circ\{\{1 \le T_0 < 10, 1 \le E_1\}, \{1 \le T_0, 1 \le E_1 < 8\}\} \\ \text{L}\lhd\text{N'}: & \text{F, E}\lhd\text{N'} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{E_1}\text{live(n)} \\ & \circ\{\{1 \le T_0 < 10, 1 \le F_1, 1 \le E_1\}, \\ & \quad \{1 \le T_0, 1 \le F_1, 1 \le E_1 < 8\}\} \end{array}$$

The accommodation of the constraints of the argument N◁N' is computed as follows:

$$\begin{array}{l} \{\{1 \le T_0 < 10, 1 \le F_1, 1 \le E_1\}, \\ \{1 \le T_0, 1 \le F_1, 1 \le E_1 < 8\}\} \\ \times\{\{E_1 = 9\}\} \times \{\{E_1 < T_2\}\} \\ = \{\{1 \le T_0 < 10, 1 \le F_1, 1 \le E_1, E_1 = 9, E_1 < T_2\}, \\ \quad \{1 \le T_0, 1 \le F_1, 1 \le E_1 < 8, E_1 = 9, E_1 < T_2\}\} \end{array}$$

The first set of constraints is satisfiable whereas the second set of constraints is unsatisfiable and thus, accordingly with definition 7, the latter is removed from the argument N◁N':

$$\begin{array}{ll} \text{N}\lhd\text{N'}: & \text{L}\lhd\text{N', P2} \Rightarrow \mathsf{Hold}^{T_0}\mathsf{Hold}^{T_2}\text{obligation} \\ & \circ\{\{1 \le T_0 < 10, 1 \le F_1, E_1 = 9, E_1 < T_2\}\} \end{array}$$

The computation shows, as expected, that the obligation holds after time 9 (after the event is trigered) from the pespectives between time 1 included and before 10, that is, before the norm n is retroactively abrogated.

10. RELATED WORK

The combination of constraints and logics is not new, with work on constraint logic programming (*e.g.*, [15, 16]) providing a natural, practical and efficient way to model and solve problems. Our representation of constrained temporal literals has been much influenced by the work presented in [7, 25]; our use of off-the-shelf constraint satisfaction technologies has also been inspired by [25]. Our approach is related with the work described in [2, 23], in that we both employ constraints to represent temporal relations and manipulate these with a view to "weaken" restrictions, thus being able to find approximate solutions. However, the machinery used in [2, 23] aims at planning systems (and associated planning formalisms), so the techniques are not directly comparable. There are also similarities between our approach and the work of [26]; however, that approach aims at manipulating norms (annotated with constraints) to avoid conflicts, aiming at a pair of norms (as opposed to a more complex structure as our arguments). Very recently, [4] combines argumentation frameworks with techniques from Soft Constraint Satisfaction Problem. The authors propose that semirings can represent "weighted" argumentation systems. This allows for developing a framework where argumentation moves have a weight and so the computation of the classical Dung's semantics has an associated weight determining the tolerance threshold of inconsistency. This proposal does not handle temporal reasoning, even though it is an interesting open problem to explore how and if [4]'s techniques can cover temporal patterns in legal argumentation.

Non-monotonicity and temporal persistence are covered by a number of different formalisms, some of which are quite popular and mostly based on variants of Event Calculus or Situation Calculus combined with non-monotonic logics (see, e.g., [22, 24]). Temporal and duration based defeasible reasoning has been also developed by [14] as an extension of Defeasible Logic.

Other approaches integrating temporal reasoning in argumentation includes Augusto and Simari's [1], Mann and Hunter's [18], and Barringer and Gabbay's [3]. [3] proposes temporal and modal languages to represent arguments in the nodes of a network and give a Kripke semantics. Closer to our approach, constraint-based temporal reasoning combining intervals and instants are integrated in an argumentation framework in [1]. Predicates as Holds, Occurs and Do are used to associate constrained temporal information to logic formulas representing properties, events and actions. Mann and Hunter in [18] encode temporal information via formulas of the form Holds(α, i) to express that α holds at interval i and propose a translation into classical propositional calculus. Compared to our work, beside differences on the semantics, none of these argumentation frameworks deal with the notions as accommodation and temporal perspectives.

Finally, in the AI&Law community, after some initial attempts using, for example, Event Calculus [19], more recently, most approaches are based on a temporal extension of Defeasible Logic [8, 9, 10, 11, 13]. Some of such extensions look quite efficient [12], but all these works are based proof-theoretic approaches to temporal reasoning in the law where no explicit temporal constraint is introduced. Even though Temporal Defeasible Logic (TDL) adopts other techniques, some basic intuitions on how arguments and time interplay in our proposal are also inspired by those contributions. However, one significant aspect regarding the notion of temporal persistence marks the difference between these two approaches. In TDL, when a persistent literal is successfully attacked, then it no longer holds after the attack stops applying, whereas in the current framework a literal, when derived, holds in the interval determined by the constraints, except when an attack applies. Accordingly, TDL captures more directly the common sense law of inertia [22]. However, there are examples where a reason r to block the persistence of a literal l is effective w.r.t. this literal only as long as r's effect persists too: when such an r no longer applies, then l comes back by inertia without any further positive reason. Quite interestingly, both treatments of temporal persistence are significant in the legal domain. The common sense law of inertia perfectly fits cases where, for instance, the termination of persistent obligations applies when they are fulfilled: the obligation to pay persists until you comply with this obligation. The second view captures quite naturally the concepts of periodicity and temporary suspension of the legal effects of norms. In any case, from a mathematical point of view, TDL and our framework can both simulate temporal suspension and the common law of inertia, respectively: the former by duplicating rules, the latter by using the negation as failure. A more accurate investigation of the relation between our system and TDL is left to future research.

11. CONCLUSIONS & FUTURE WORK

In this paper we explored temporal issues of norms and arguments. More specifically, we addressed conditional norms which come into force during periods of time. We proposed an extension of arguments in which temporal aspects are factored in – we employed constraints to capture a simple concept of chronology (or temporal ordering); we used off-the-shelf constraint satisfaction techniques to manipulate these constraints. We have observed that constraints provide an elegant way to handle conflicts (or attacks) between arguments: by adding constraints to a defeated argument, we ensured it is no longer rebutted – we said such an argument has been accommodated. We sketched a fixed-point semantics for our argumentation framework and presented a dialogue game to check if an (accommodated) argument belongs to the grounded extension of our semantics. We showed how it can model various significant legal scenarios.

We would like to extend our work in several directions. The negation as failure is under investigation in order to account for undercutting attack and associated temporal aspects. An immediate and useful extension is to use constraints not just to represent temporal issues, but to also restrict the value of variables in predicates; our literals would be first-order atomic predicates with associated constraints. In the same direction, constraints can also be used to deal with spatial aspects in legal reasoning.

12. REFERENCES

[1] J. Augusto and G. Simari. Temporal defeasible reasoning. *Knowledge and Information Systems*, 3(3):287–318, 2001.

[2] J. A. Baier, F. Bacchus, and S. A. McIlraith. A heuristic search approach to planning with temporally extended preferences. *Artificial Intelligence*, 173(5-6):593–618, 2009.

[3] H. Barringer and D. M. Gabbay. Time for verification. chapter Modal and temporal argumentation networks, pages 1–25. Springer-Verlag, Berlin, Heidelberg, 2010.

[4] S. Bistarelli and F. Santini. A common computational framework for semiring-based argumentation systems. In *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 131–136, Amsterdam, 2010. IOS Press.

[5] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57:9:1–9:41, February 2010.

[6] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[7] A. García-Camino, J. Rodríguez-Aguilar, C. Sierra, and W. Vasconcelos. Constraint rule-based programming of norms for electronic institutions. *Autonomous Agents and Multi-Agent Systems*, 18:186–217, 2009. 10.1007/s10458-008-9059-4.

[8] G. Governatori, J. Hulstijn, R. Riveret, and A. Rotolo. Characterising deadlines in temporal modal defeasible logic. In *Australian Conference on Artificial Intelligence*, pages 486–496, 2007.

[9] G. Governatori, M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Norm modifications in defeasible logic. In *Proceedings of the 18th International Conference on Legal Knowledge and Information Systems*, pages 13–22. IOS Press, 2005.

[10] G. Governatori, M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Variants of temporal defeasible logic for modelling norm modifications. In *11th International Conference on Artificial Intelligence and Law (ICAIL 2007)*, pages 155–159. ACM Press, 2007.

[11] G. Governatori and A. Rotolo. Changing legal systems: Legal abrogations and annulments in defeasible logic. *The Logic Journal of IGPL*, 2010.

[12] G. Governatori and A. Rotolo. On the complexity of temporal defeasible logic. In *13 International Workshop on Non-Monotonic Reasoning (NMR 2010)*, 2010.

[13] G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, pages 25–34. ACM Press, 2005.

[14] G. Governatori and P. Terenziani. Temporal extensions to defeasible logic. In M. Orgun and J. Thornton, editors, *20th Australian Joint Conference on Artificial Intelligence*, pages 476–485, 2007.

[15] J. Jaffar and M. J. Maher. Constraint Logic Programming: A Survey. *Journal of Logic Progr.*, 19/20:503–581, 1994.

[16] J. Jaffar, M. J. Maher, K. Marriott, and P. J. Stuckey. The Semantics of Constraint Logic Programs. *Journal of Logic Programming*, 37(1-3):1–46, 1998.

[17] H. Kelsen. *General theory of norms*. Clarendon, Oxford, 1991.

[18] N. Mann and A. Hunter. Argumentation using temporal knowledge. In *Proceeding of the 2008 conference on Computational Models of Argument: Proceedings of COMMA 2008*, pages 204–215, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[19] R. H. Marín and G. Sartor. Time and norms: a formalisation in the event-calculus. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, pages 90–99. ACM Press, 1999.

[20] H. Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *Proceedings of the 10th international conference on Artificial intelligence and law*, ICAIL '05, pages 85–94, New York, NY, USA, 2005. ACM.

[21] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1), 1997.

[22] M. Shanahan. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press, Cambridge, MA, 1997.

[23] S. Sohrabi, J. Baier, and S. A. McIlraith. HTN planning with preferences. In *Twenty First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1790–1797, Pasadena, California, 2009.

[24] H. Turner. Representing actions in logic programs and default theories: A situation calculus approach. *Journal of Logic Programming*, 31(1-3):245–298, 1997.

[25] W. Vasconcelos, M. Kollingbaum, and T. Norman. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19:124–152, 2009. 10.1007/s10458-008-9070-9.

[26] W. Vasconcelos and T. Norman. Contract formation through preemptive normative conflict resolution. In *Procs. 12th Int'l Conf. of the Catalan Assoc. for A. I.*, pages 179–188, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.