Variants of Temporal Defeasible Logics for Modelling Norm Modifications

Guido Governatori[1]*, Antonino Rotolo[2]†, Régis Riveret[2]†, Monica Palmirani[2]†, Giovanni Sartor[2]†
[1] School of ITEE, The University of Queensland, Brisbane, Australia
[2]CIRSFID and Law Faculty, University of Bologna, Italy
guido@itee.uq.edu.au,{rotolo, rriveret, palmiran,sartor}@cirsfid.unibo.it

ABSTRACT

This paper proposes some variants of Temporal Defeasible Logic (TDL) to reason about normative modifications. These variants make it possible to differentiate cases in which, for example, modifications at some time change legal rules but their conclusions persist afterwards from cases where also their conclusions are blocked.

1. BACKGROUND AND MOTIVATION

This paper presents some variants of Temporal Defeasible Logic (TDL) [5, 4] to reason about different aspects of norm modifications. The issues discussed in this paper are similar to those assumed in [5], namely: (1) to reason about conditional modifications of conditional normative provisions, where the former ones apply under, and are conditioned to, the occurrence of some uncertain events; (2) to identify criteria for detecting and solving conflicts between textual modifications; (3) to clarify the specific role played by the temporal dimension in modelling norm modification processes. The novelty of this work is that we are interested in defining different temporal constraints according to which the elements of a normative system, and the conclusions that follow from them, can, or cannot persist over time. Indeed, several options are available, and each of them corresponds to a specific way through which norm modifications can take place and behave.

Norm applications and modifications take place along the axis of time. In particular, a rule is represented at least as $(a^t \Rightarrow b^{t'}) : t''$, where instants t and t' indicate the time at which a and b hold, while t'' is the time when the rule is in force. A temporal model should allow us to give an accurate account of the dynamics of norms and therefore to manage legal modifications consistently with legal principles [10].

A legal system is defined as a set of documents fixed at a time t and which have been issued by an authority and whose validity depends on rules that determine, for any given time, whether a single document belongs to the system. Formally:

$$LS(t) = D_1(t), D_2(t), D_3(t), \ldots, D_m(t)$$

where $m \in \mathbb{N}$, D_i denotes documents and t is a fixed time in a discrete representation. A normative system, in turn, takes the documents belonging to a legal system and organises them to reflect their evolution over time. A normative system should therefore be defined as a particular discrete time-series of legal systems that evolves over time. In formal terms:

$$NS = LS(t_1), LS(t_2), LS(t_3), \ldots, LS(t_j),\ j \in \mathbb{N}.$$

The passage from a legal system to another legal system is effected either by normative modifications or by simple persistence. However, norm-modifying provisions, too, belong to legal systems, and, as other legal provisions, they can be seen as conditional statements. Accordingly, also modifications have three temporal dimensions, these being attached to the conditions, to the effects, and to the overall conditional. These temporal dimensions refer to the efficacy, applicability, and force of the provision, respectively. Furthermore, one has to consider the time of observability of the normative system. Consider a law r of 2001 nullified in 2005: the change affects the entire normative system because the legal text is removed from the system as if it had never been there in the first place (ex-tunc removal). The same would happen, e.g., with a temporary law decree that does not pass into law or with a retroactive abrogation [7]. The peculiarity of system changes shows up when we query the system to retrieve information from it: if today (e.g., 2007) we ask for all the laws in force in 2001, law r will not turn up and the system will look as if that law had never been in force in the first place. But if we enter the query as if we were in 2001, when the annulment had not yet occurred, law r will show up as being in force, and the entire system will reflect that fact. This difference depends on the temporal point of view from which we query the system and this refers to the time of observability of a normative system.

It is possible to identify different kinds of normative conditional in a temporal setting, and so different ways of how normative modifications temporally affect the normative system [6, 5]. For example, we can distinguish between *persistent* and *transient* rules: the former, if applicable, permit to infer literals that persist unless some other, subsequent, and incompatible events or states of affairs terminate them; the latter allow for the inference of literals which hold on the condition and only while the antecedents of these rules hold. But these characterisation of transiency and persistency provides only a partial picture of how things can or cannot evolve over time. Persistency is a notion which need to be linked with a specific temporal perspective. Consider the following legal provisions

$$r_1 : (a^{10} \Rightarrow b^{10}) : 10 \qquad r_2 : (b^{10} \Rightarrow c^{10}) : 10$$

*Supported by the Australian Research Council under the Discovery Project DP0558854.
†Supported by the European project for Standardized Transparent Representations in order to Extend Legal Accessibility (ESTRELLA, IST-4-027655).

ICAIL '07, June 4-8, 2007, Palo Alto, CA USA.
Copyright 2007 ACM 978-1-59593-680-6 ...$5.00.

Both r_1 and r_2 are assumed to produce persistent conclusions, namely, that b and c persist after 10 if r_1 and r_2 are applicable at 10. Now suppose that a modification m at time 20 applies to r_1, holding 10, and nullifies it. This is a case of retroactive modification, as the norm-modifying provision m is in force at 20. If we obtain a at 10, this makes r_2 applicable, thus deriving c at 10 and afterwards. So, despite, the annulment of r_1, its legal effects are propagated. A simple solution to obviate the problem is to state that the conclusion of r_1 can be propagated after a certain time only if r_1 has not been in the meantime nullified. This is not enough, as this would simply block b from 20 afterwards, but it would not apply to c, which was derived at 10: c will hold after 20 independently of m. Of course, we can imagine scenarios in which the applicability of r_1 may support a large number of rule chains, thus posing a serious computational problem. Indeed, this is a concrete difficulty which arises when we aim to model ex-tunc modifications which are meant to cancel *all* legal effects of a certain legal provision.

But things are even more complex. In fact, persistency and transiency can apply not only to conclusions of rules, but also to the rules themselves, to the time of observability, and also to derivations (i.e., queries). This gives rise to several options regarding how modifications affect the legal system over time. We will consider the interplay among different time-lines; in particular, among the time-lines through which *conclusions*, *derivations*, and *rules* persist either within a certain version of the legal system or across different temporal versions of the legal system.

We will also distinguish between *modifiable* and *non-modifiable* rules. Non-modifiable normative provisions are important when constitutions are considered. In fact, in many legal systems some parts of the constitution cannot be changed even by the special procedures the constitution sets forth for its revision.

2. TEMPORAL DEFEASIBLE LOGIC

Defeasible Logic (DL) [9, 1] is a simple, efficient (indeed the complexity is linear [8]) but flexible non-monotonic formalism capable of dealing with many different intuitions of non-monotonic reasoning with a logic programing like language. DL is characterised by an argumentation semantics [3] and also has several efficient implementations [2]. Temporal Defeasible Logic (TDL) is an umbrella expression to designate extensions of DL to capture time. TDL has proved useful in modelling temporal aspects of normative reasoning, such as temporalised normative positions [6]; in addition, the notion of a temporal viewpoint –the temporal position from which things are viewed– allows for a logical account of norm modifications and retroactive rules [5]. Here we present some variants that deal with temporal dimensions as exposed in Section 1. Temporal aspects are integrated by two means: first by introducing temporal coordinates and second normative modifications.

[6] extended DL with temporalised literals, i.e., every literal in the logic has associated to it a timestamp. Thus we have expressions of the type $a : t$, meaning that a holds at time t. This means that we have to give the condition to prove a literal at time t. So we have to consider whether a conclusion is transient (holding at precisely one instant or time) or whether it is persistent. To prove that a holds at t, we can prove that a held at a previous instant t' and then for all instant in between t and t', it is not possible to terminate a. We will refer to this property as *persistence of a conclusion*.

As we have argued above not only literals (propositions) have their temporal validity, but this is true for the other components of our knowledge: we can speak of the time of force of a rule, i.e., the time when a rule can be used to derive a conclusion given a set premises. In this perspective we can have expressions like $r : (a^{t_a} \rightarrow b^{t_b}) : t_r$, meaning that the rule r is in force at time t_r, or in other words, we can use the rule to derive the conclusion at time t_r. The full semantics of this expression is that at time t_r we can derive that b holds at time t_b if we can prove that a holds at time t_a. But now we are doing a derivation at time t_r, so the conclusion b^{t_b} is derived at time t_r and the premise a^{t_a} must be derived at time t_r as well. In the same way a conclusion can persist, and we can have the same for rules and then for derivations.

What we derive depends on what rules are valid, and on the normative content of rules, at the time when we do the derivation. In addition, we have to consider the case that the content of a rule can be changed. Thus we have to devise a mechanism to capture this phenomenon. To this end we introduce meta-rules, i.e., rules where the consequent is itself a rule and not only a simple proposition. In addition to keep track of the changes to a norm, i.e., to represent a normative systems as defined in Section 1, we introduce the notion of a repository, i.e., a snap-shot of rules and literals known to exist at a specific time instant. In the rest of the section we will give a formal presentation of the notions discussed so far.

The language of TDL is based on a (numerable) set of atomic proposition $Prop = \{p, q, \ldots\}$, a set of rule labels $\{r_1, r_2, \ldots\}$, a discrete totally ordered set of instants of time $\mathscr{T} = \{t_1, t_2, \ldots\}$, and the negation sign $\neg$. A plain literal is either an atomic proposition or the negation of it. Given a literal l with $\sim l$ we denote the complement of l, that is, if l is a positive literal p then $\sim l = \neg p$, and if $l = \neg p$ then $\sim l = p$. If l is a literal and t is an instant of time, i.e., $t \in \mathscr{T}$, the l^t is a temporal literal. The meaning of a temporal literal l^t is that l holds at time t.

Knowledge in DL can be represented in two ways: facts and rules. *Facts* are indisputable statements, represented either in form of states of affairs and actions that have been performed (literals). For example, “John is a minor in year 2007”. In the logic, this might be expressed as $Minor(John)^{2007}$. A *rule* is a relation between a set of of premises and a conclusion, where the admissible conclusions are either literals or rules themselves, and the conclusions and the premises will be qualified with the time when they hold. We consider two classes of rules: *meta-rules* and *proper rules*. Meta-rules describe the inference mechanism of the institution on which norms are formalised and can be used to establish conditions for the creation and modification of other rules or norms, while proper rules corresponds to norms in a normative systems. Rules can also be partitioned according to their strength into *strict rules* (denoted by $\rightarrow$), *defeasible rules* (denoted by $\Rightarrow$) and defeaters (denoted by $\leadsto$). Strict rules are rules in the classical sense: they are monotonic and whenever the premises are indisputable so is the conclusion. Defeasible rules, on the other hand, are non-monotonic: they can be defeated by contrary evidence. Finally defeaters are the weakest rules: they do not support conclusions, but can be used to block the derivation of opposite conclusions.

We define the set of rules as follows: a rule is either a meta-rule or a proper rule or the empty rule $\perp$, where

- If r is a proper rule then $\sim r$ is a rule.
- If r is a rule and $t \in \mathscr{T}$, then $r : t$ is a temporalised rule.
- If r is a temporalised rule and $t \in \mathscr{T}$, then $r@t$ is a temporalised rule with viewpoint.
- Let A be a finite set of temporal literals, C be a temporal literal and r a temporalised rule, then $A \hookrightarrow C$ is a rule, and $A \hookrightarrow r$ and $A \hookrightarrow \sim r$ are meta-rules (henceforth we use $\hookrightarrow$ as a meta-variable for either $\rightarrow$ when the rule is a strict rule, $\Rightarrow$ when the rule is a defeasible rule, and $\leadsto$ when the rule is a defeater).

Given a set R of rules, we denote the set of all strict rules in R by R_s, the set of defeasible rules in R by R_d, the set of strict and defeasible rules in R by R_{sd}, and the set of defeaters in R by R_{dft}. $R[q]$ denotes the set of rules in R with consequent q. $A(r)$ denotes the antecedent or body of r and $C(r)$ the head or consequent of r.

Norms in a normative system have two temporal dimensions: when the norm is in force in it, and when the norm exists in the normative system from a certain viewpoint. Temporalised rules capture only one dimension, the time of force. For the other dimension we introduce the notion of temporalised rule with viewpoint. A temporalised rule with viewpoint is a function $\mathscr{T} \mapsto (\mathscr{T} \mapsto \text{Rules})$. The above inductive definition makes it possible to have nested rules. For example, the following is a meta-rule with viewpoint:

$$((p^{t_p}, q^{t_q} \Rightarrow (p^{t_p} \Rightarrow s^{t_s}) : t_v)) : t'_v @ t_r \quad (1)$$

(1) exists from viewpoint t_r, is in force from t'_v and means that if p is true at t_p and q at t_q, then the rule $p^{t_p} \Rightarrow s^{t_s}$ is in force at t_r.

A normative system NS is a sequence of legal systems LS (see Section 1). The temporal viewpoint corresponds to a legal system while the temporal dimension temporalising a rule corresponds to the time-line inside a legal system. Thus the meaning of an expression $r : t_v @ t_r$ is that we take the value of the temporalised rule $r : t_v$ in the legal systems $LS(t_r)$. Accordingly, a legal system is just a repository (set) of norms (implemented as temporal functions).

We extend the notion of viewpoint to temporalised literals: if l^t is a temporal literal and $t' \in \mathscr{T}$, then $l^t @ t'$ is a *fully temporalised literal*. The meaning of $l^t @ t'$ is that it is possible to use the information that l holds at t from time t', i.e. that the information in available in the repository corresponding to time t'.

Finally, for every literal and rule and every temporal dimension we have to specify whether the element is persistent or transient for that temporal dimension. The interpretation of transient and persistent elements is as follows: A transient temporalised literal $l^{t,trans}$, means that l holds at time t, while a persistent temporal literal $l^{t,pers}$ signals that l holds for all instants of time after t (t included), for the time-line of the legal system in which the literal is found. For a transient fully temporalised literal $l^t @ (t', trans)$ the reading is that the validity of l at t is specific to the legal system corresponding to repository associated to t', while $l^t @ (t', pers)$ indicates that the validity of l at t is preserved when we move to legal systems after the legal system identified by t'. An expression $r : (t, trans)$ sets the value of r at time t and just at that time, while $r : (t, pers)$ sets the values of r to a particular instance for all time after t (t included). These two notions refer to the time-line of a specific legal system, and a similar reading can be given for persistence for the time-line of the normative system[1].

A normative system is represented by a temporalised defeasible theory, which is a structure

$$(\mathscr{T}, F, R^{\text{nm}}, R^{\text{meta}}, R^{\text{mod}}, \prec)$$

where $\mathscr{T}$ is a totally ordered discrete set of time points, F is a finite set of facts, where a fact is a fully temporalised literals, R^{nm} is a finite set of unmodifiable rules, R^{meta} is a finite set of meta rules, R^{mod} is a finite set of proper rules, and $\prec$, the superiority relation over rules is formally defined as $\mathscr{T} \mapsto (\mathscr{T} \mapsto \text{Rules} \times \text{Rules})$.

The superiority relation $\prec$ determines the relative strength for rules for every instant in every legal systems. Thus it is possible that a rule r is both stronger and weaker than another rule s in a legal system, and then that two rules in different legal systems in a normative system have opposite relative strengths.

We are now ready to define how conclusions can be obtained in TDL. Notice that every time we have to use a rule, we have to ensure that the rule is derivable from the theory. Proof conditions for rules are slightly different from those for literals (though they follow the same intuition). Accordingly, we will give separate proof conditions for deriving literals and for deriving rules. In addition we have to extend the notion of complement to cover rules. Here, we use again the intuition that a rule is a function. Given a rule instance $r : A(r) \hookrightarrow b : t$,

$$R[\sim r : t] = \{r : \bot : t\} \cup \{r : A'(r) \hookrightarrow b' : t | A'(r) \neq A(r) \text{ or } b' \neq b\}$$

The main notion at hand is the notion of derivation (or proof). A *proof* P is a finite sequence of tagged expressions such that: (1) Each expression is either a temporalised rule or a temporalised literal; (2) Each tag is one the following: $+\Delta t @ t'$, $-\Delta t @ t'$, $+\partial t @ t'$, $-\partial t @ t'$; (3) The proof conditions *strict rule provability*, *defeasible rule provability*, *strict literal provability* and *defeasible literal provability* given below are satisfied by the sequence P.

Given a proof P we use $P(n)$ to denote the n-th element of the sequence, and $P[1..n]$ denotes the first n elements of P.

A proof tag has four components: (1) sign, (2) tag, (3) derivation time and (4) repository time. Accordingly, the meaning of the proof tags is a follows:

- $\pm\Delta t @ t'\ x^{t_x}$ (resp. $\pm\Delta t @ t'\ r : t_r$) meaning that we have (we do not have) a definite derivation of x^{t_x} (resp. $r : t_v$) at time t using the elements in the repository at time t';
- $\pm\partial t @ t'\ x^{t_x}$ (resp. $\pm\partial t @ t'\ r : t_r$) meaning that we have (we do not have) a defeasible derivation of x^{t_x} (resp. $r : t_v$) at time t using the elements in the repository at time t'.

We will adopt in the proof conditions the following convention for the various times involved: t_d is the time with respect to which we do the derivation and it refers to the time-line within a legal system, t_r is the repository time, thus it is the time-line of the normative system. Finally, the last temporal dimension is the object time, which for a rule is the time of force t_v, for a literal a it is the time when the literal holds; we use a^{t_a} for a temporal literal. The derivation and the repository times are parameters of the proof tags.

The mechanism for a derivation in the framework is as follows. A derivation corresponds to a query, and the query is parametrised by two temporal values: the repository time and the derivation time. The repository time is used to time-slice the information relevant for the query using the time-line of the normative system. This means that we retrieve all elements of the theory where the repository time is equal to the repository time of the query and all elements whose repository time is less than the repository time of the query but the elements carry over due to persistence over repositories. After this step we have the legal system in force at the repository time. At this stage the derivation time kicks in. Similarly to what we have done in the previous step, we use the value of the derivation time to time-slice the legal system under analysis. In particular we consider all rules whose time of force is equal to the derivation time, or rules whose time of force precedes the current derivation time but carries over to it because such rules are marked as persistent. Finally, we consider the temporalised literals in the rules resulting from the previous steps, and we check if the literals are provable with the time with which they appear in the rules.

Strict Rule Provability

If $P(n+1) = +\Delta t_d @ t_r\ r : t_v$ then
1) $r : t'_v @ t'_r \in R^{\text{nm}}$ or
2) $\exists s @ t'_r \in R^{\text{meta}}_s : \forall a^{t_a} \in A(s), +\Delta t_d @ t_r\ a^{t_a} \in P[1..n]$, or
3) $+\Delta t'_d @ t''_r\ r : t'_v$.

where: (1) if r is persistent, then $t'_v \leq t_v$; (2) if r is transient, then $t_v = t'_v$; (3) if facts, rules and meta-rules are persistent across repositories, then $t'_r < t_r$, otherwise $t'_r = t_r$; (4) $t'_d < t_d$ if conclusions are

[1] To simplify the presentation we specify if an element is persistent or transient only when relevant.

persistent within a repository; (5) $t''_r < t_r$ if conclusions are persistent across repositories.

Notice that for clause (2) we must be able to prove the antecedent of the meta-rule s with exactly the same reference point, i.e., combination of derivation time t_d and repository time t_r as the reference point of the conclusion we prove, i.e., $r : t_v$; whether the literals used to apply s are obtained by persistency or by a direct derivation with the appropriate time reference depends on the proof conditions for literals and the variant of temporal defeasible logic at hand. Finally clause (3) is the persistence clause for strict derivation of rules.

Defeasible Rule Provability

If $P(n+1) = +\partial t_d@t_r\ r : t_v$, then
1) $+\Delta t_d@t_r\ r : t_v$ or
2) $-\Delta t_d@t_r \sim r : t_v$ and
2.1a) $r : t'_v@t'_r \in R^{\text{mod}}$ or
2.1b) $\exists s : t_s \in R^{\text{meta}}_{sd}[r : t'_v] : \forall a^{t_a} \in A(s), +\partial t'_d@t''_r\ a^{t_a} \in P[1..n]$ and
2.2) $\forall m : t_m \in R[\sim r : t_v]$ either
.1) $\exists b : t_b \in A(m) : -\partial t''_d@t'''_r\ b^{t_b} \in P[1..n]$ or
.2) $m : t_m \prec^{t_r}_{t_d} r : t_r$, if 2.1a obtains or
.3) $m : t_m \prec^{t_r}_{t_d} s : t_s$, if 2.1b obtains or
.4) $\exists w : t_w \in R[r : t''_v] : \forall c^{t_c} \in A(w), +\partial t'''_d@t''''_r\ c^{t_c} \in P[1..n]$ and $m : t_m \prec^{t_r}_{t_d} w : t_w$

where (1) if r is persistent, then $t'_v \leq t_v$; (2) if r is transient, then $t_v = t'_v$; (3) if a^{t_a}, (resp. b^{t_b}, c^{t_c}) is persistent within the repository at t_r, then $t'_d \leq t_d$ (resp. $t''_d \leq t_d$, $t'''_d \leq t_d$); (4) if a^{t_a} (resp. b^{t_b}, c^{t_c}) is transient within the repository at t_r, then $t'_d = t_d$) (resp. $t''_d = t_d$, $t'''_d = t_d$); (5) if a^{t_a}'s, b^{t_b}'s and c^{t_c}'s are persistent with respect to repositories (i.e., conclusions are persistent), then $t''_r, t'''_r, t''''_r \leq t_r$; (6) if $r : t'_v$ and s (i.e., facts, rules, and meta-rules) are persistent with respect to repositories, then $t'_r \leq t_r$.

Strict Literal Provability

If $P(n+1) = +\Delta t_d@t_r\ p^{t_p}$, then
1) $p^{t'_p}@t'_r \in F$; or
2) $\exists r : t_v \in R_s[p^{t'_p}]$ such that
.1) $+\Delta t_d@t_r\ r : t'_v \in P[1..n]$, where $t'_v = t_d$ and
.2) $\forall a^{t_a} \in A(r) : +\Delta t_d@t_r\ a^{t_a} \in P[1..n]$; or
3) $+\Delta t'_d@t'_r\ p^{t'_p} \in P[1..n]$.

where: (1) if p is persistent, then $t'_p \leq t_p$; (2) if p is transient, then $t'_p = t_p$; (3) if r is persistent, then $t'_v \leq t_v$; (4) if r is transient, then $t_v = t'_v$; (5) if facts, rules and meta-rules are persistent across repositories, then $t'_r < t_r$, otherwise $t'_r = t_r$; (6) if conclusions are persistent within a repository, then$t'_d < t_d$; (7) if conclusions are persistent across repositories, then$t'_r < t_r$.

Defeasible Literal Provability

If $P(n+1) = +\partial t_d@t_r\ p^{t_p}$, then
1) $+\Delta t_d@t_r\ p^{t_p} \in P[1..n]$ or
2) $-\Delta t_d@t_r \sim p^{t_p} \in P[1..n]$ and
2.1) $\exists r : t_v \in R_{sd}[p^{t'_p}]$ such that
$+\partial t'_d@t'_r\ r : t'_v \in P[1..n]$ and
$\forall a^{t_a} \in A(r), +\partial t'_d@t'_r\ a^{t_a} \in P[1..n]$, and
2.2) $\forall s : t_s \in R[\sim p^{t_{\sim p}}]$ if $+\partial t''_d@t''_r s : t'_s \in P[1..n]$, then either
.1) $\exists b^{t_b} \in A(s), -\partial t''_d@t''_r\ b^{t_b} \in P[1..n]$ or
.2) $\exists w : t_w \in R[p : t''_p]$ such that
$+\partial t''_d@t''_r w : t_w \in P[1..n]$ and
$\forall c^{t_c} \in A(w), +\partial t''_d@t''_r\ c^{t_c} \in P[1..n]$ and $s : t_s \prec^{t_r}_{t_d} w : t_w$.

where (1) if p is persistent, $t'_p \leq t_{\sim p} \leq t_p$, otherwise $t'_p = t_{\sim p} = t_p$; (2) $t'_s \leq t_v$, if s is persistent, otherwise $t_s = t'_s = t_v$; (3) $t_d \leq t'_s$, if s is persistent, otherwise $t_s = t'_s = t_d$; (4) if conclusions are persistent over derivations (i.e., $+\partial t'_d@t_r p^{p_t}$ implies $+\partial t_d@t_r p^{p_t}$ where $t'_d < t_d$), then, $t'_d \leq t''_d \leq t_d$; (5) if conclusions are persistent over repositories, then $t'_r \leq t''_r \leq t_r$.

The above proof conditions produce classes of TDLs, according to the conditions on the temporal parameters. In particular it is possible to define variants capturing different types of persistence. For our purpose we mention *rule* and *causal conclusion* persistence.

Generally once a norm has been introduced in a normative system, or better in a specific legal system of the normative system, the norm continues to be in the normative system unless it is explicitly removed (see Section 3 for some possible types of removal). This means that the norm must be included in all legal systems succeeding the legal system in which it has been first introduced. This effect is achieved by specifying that the derivation of rules is persistent over repositories.

If we can prove a conclusion with respect to a specific legal system in some cases we have to propagate it to successive legal systems. In particular this is the case when we have causal conclusions. This was the option we explored in [5]. However, for some type of norm modifications, namely annulment (see Section 3), we have to block the persistence of conclusions over repositories when the reasons for deriving a conclusion are no longer in the legal system. This effect depends on whether derivations of conclusions are persistent over repositories, and it is in function of the particular type of modification we want to implement. Consider:

$$r : a^{10} \Rightarrow b^{(20,pers)} : 10@(1,trans) \qquad s : b^{30} \Rightarrow c^{(30,pers)} : 15@(1,pers)$$

Since r is marked as transient, the rule can be used only in repository 1, while s can be used in all repositories after repository 1.Given $a^{10}@1$ we can first derive $+\partial 10 : 1\ b^{(20,pers)}$. Since b is persistent we have $+\partial 10@1\ b^{20}$. The second rule cannot be applicable, since its validity time is 15; to apply it we have to assume that derivations are persistent within a repository. If this is the case then we obtain $+\partial 15 : 1\ b^{20}$, which then makes rule s applicable, and from which we get $+\partial 15 : 1\ c^{30}$. If we have that conclusions are persistent across repositories, then we can conclude $+\partial 15 : 2\ c^{30}$. Notice that we can conclude $+\partial 15@2\ c^{30}$ even if the reasons for deriving it (i.e., rule r) do not persist across repositories.

3. NORM MODIFICATIONS IN TDL

We consider four kinds of modifications: substitution (which replaces some textual components of a provision with other textual components, or a provision with another provision), derogation (the derogating provision limits the effects of the derogated provision), annulment (which cancels *ex tunc* a provision and prevents it to produce any normative effect), and abrogation (which cancels a provision but does not cancel the effects that were obtained before the modification). The application of a norm-modifying provision changing a rule r is represented by deriving a set MOD of rules that change the status, or even single parts, of r. This typically happens via the application of one or more meta-rules that lead to obtain the rules in MOD. In many cases (but derogation is an exception), modifications imply that MOD should include a new version of r, namely, that a rule, labelled by r, is in MOD but it is different from the r which was in force before deriving MOD.

The following reasoning patterns correspond to the mentioned types of modification.

Substitution

Preconditions: $+\partial t_d@t_r\ r : (A \hookrightarrow C) : t_v$;
Derived rules MOD: $+\partial t'_d@t'_r\ r : (A' \hookrightarrow C') : t'_v$;

Constraints: (1) $A' \neq A$ or $C' \neq C$, and (2) $t'_r \geq t_r$ and $t'_v \neq t_v$, and (3) $-\Delta t_d @ t_r\ r : (A \hookrightarrow C) : t_v$.

Derogation

Preconditions: $+\partial t_d @ t_r\ r : (A \Rightarrow C) : t_v$;
Derived rules MOD: $+\partial t'_d @ t'_r\ r' : (A' \Rightarrow C') : t'_v$, $+\partial t'_d @ t'_r\ r'' : (A' \leadsto \sim C) : t'_v$
Constraints: (1) $A \subset A'$ and $C' \neq C$, and (2) $t'_r \geq t_r$ and $t'_v \neq t_v$, and (3) $-\Delta t_d @ t_r\ r : (A \Rightarrow C) : t_v$.

Annulment and Abrogation

Preconditions: $+\partial t_d @ t_r\ r : (A \hookrightarrow C) : t_v$;
Derived rules MOD: $+\partial t'_d @ t'_r\ (r : \bot) : t'_v$;
Constraints: (1) $t'_r \geq t_r$, $t'_v \neq t_v$, and (2) $-\Delta t_d @ t_r\ r : (A \hookrightarrow C) : t_v$.

The basic assumptions for all cases are that a modification can be applied to r only if r is modifiable and it exists. The first assumption is captured by the last constraint in each case. Let us consider rule existence. Minimally, the existence of rule r is a notion relative to repositories, thus the modification should take place in a subsequent or in the same temporal repository in which r exists. An additional constraint, which has not been mentioned here, is that the modification (i.e., typically, meta-rules) should be in force at a time subsequent to the time when r is in force. This requirement, which is adopted in many cases in legal systems, states in fact that we cannot modify a normative provision which is not yet in force. But this is not general necessary, as we can imagine a situation in which a normative provision r is issued at t but will start to be in force only from $t+n$. In the time span from t to $t+n$ an authority could change r even if it is not yet in force. Consider, for example, a norm r issued in 2007 stating that 60 years people can no longer retire. r will be in force in 2009, and then immediately applicable and effective. Suppose that the new government deliberates in 2008 to remove r. This is indeed possible, even if r is not yet in force.

Substitution is the modification that changes, partially or entirely, the antecedent or the consequent (or both) of a normative provision r. *Derogation* can be applied only to defeasible rules: if applied to r, it permits to introduce an exception r' of r; in this case, the derivation of the exception r' is drawn together with inferring a defeater r'' that blocks the derivation of the direct effect of r when r' is applicable. *Abrogation* and *annulment* of r have basically the same structure: what makes the former different from the latter is the treatment of the effects of the modified rule. As we mentioned, annulment cancels all effects of r, whereas abrogation does not. Annulment is thus obtained by blocking persistency of derivations across repositories. The conclusions of the annulled rule will only be derived in the repository where the modification does not occur. We have now to see when norm modifications can be in conflict.

Conflicting Modifications	
annul$(r : t_v) : t_a @ t$	subst$(r : t_v) : t'_a @ t'$
abrog$(r : t_v) : t_a @ t$	subst$(r : t_v) : t'_a @ t'$
annul$(r : t_v) : t_a @ t$	derog$(r : t_v) : t'_a @ t'$
abrog$(r : t_v) : t_a @ t$	derog$(r : t_v) : t'_a @ t'$
subst$(r : t_v) : t_a @ t$	derog$(r : t_v) : t'_a @ t'$

The previous table summarises the basic conflicts between the norm modifications we have considered. To save space, we refer to previous inference patterns when we need to specify rules r' and r'' for derogation: r' is the rule which corresponds to the exception of r, while r'' is the defeater that blocks the conclusion of r in the exceptional case (see above). Notice that in all cases a conflict obtains only if the conflicting modifications apply to the same time instant and in the same repository, i.e., $t_a = t'_a$ and $t = t'$. Annulment and abrogation of r are incompatible with any substitution in r (first two rows from top). A similar intuition holds for the two subsequent rows: it is impossible to derogate to r if this rule is dropped from the system. Finally, the cases in the last row from the bottom state that a substitution in r is incompatible with a derogation if at least one literal used in r' or r'' to derogate to r is replaced in r' or r'', formally, $A(r' : t'_v) \cap A(r : t_v) \neq A(r : t_v)$ or $C(r'' : t'_v) \neq \sim C(r : t_v)$.

4. CONCLUSIONS

We extended the logic of [5] to capture different temporal aspects of the norm-modification process. This extension increases the expressive power of the logic and it allows us to represent meta-norms describing norm-modifications by referring to a variety of possible time-lines through which conclusions, rules and derivations can persist over time. We outlined the inferential mechanism needed for the derivation of rules and literals. In particular, we identified several temporal constraints that permit to allow for, or block, persistency with respect to specific time-lines. This virtually produces different variants of TDL according to whether a condition is adopted.

We described some issues related to norm modifications and versioning and we illustrated the techniques with some relevant modifications such as annulment, abrogation, substitution and derogation. In particular, we solved the problem of how legal effects of *ex-tunc* modifications, such as annulment, can be blocked after the modification applied. The idea we suggested is to block persistency of derivations across repositories. In other words, the conclusions of the annulled rule will only be derived in the repository in which the modification does not occur.

The proposed methodology illustrates the possibilities of the formalism and we intend to apply it to the logical analysis of a larger corpus of norm-modifications.

5. REFERENCES

[1] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2(2):255–287, 2001.

[2] N. Bassiliades, G. Antoniou, and I. Vlahavas. DR-DEVICE: A defeasible logic system for the Semantic Web. In *2nd PPSWR*, Springer, 2004.

[3] G. Governatori, M. J. Maher, D. Billington, and G. Antoniou. Argumentation semantics for defeasible logics. *Journal of Logic and Computation*, 14(5):675–702, 2004.

[4] G. Governatori, V. Padmanabhan, and A. Rotolo. . In *9th PRICAI*, LNAI 4099, 31–40. Springer, 2006.

[5] G. Governatori, M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Norm modifications in defeasible logic. In *JURIX 2005*, 13–22. IOS Press, 2005.

[6] G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *10th ICAIL*, 25–34. ACM Press, 2005.

[7] R. Guastini. *Teoria e dogmatica delle fonti*. Giuffré, 1998.

[8] M. J. Maher. Propositional defeasible logic has linear complexity. *Theory and Practice of Logic Programming*, (6):691–711, 2001.

[9] D. Nute. Defeasible reasoning. In *20th HICSS*, 470–477. IEEE press, 1987.

[10] M. Palmirani and R. Brighi. Time model for managing the dynamic of normative system. In *EGOV 2006*, 207–218, Springer, 2006.