Changing Legal Systems: Abrogation and Annulment Part II: Temporalised Defeasible Logic

Guido Governatori[1] and Antonino Rotolo[2]

[1] National ICT Australia, guido.governatori@nicta.com.au
[2] CIRSFID/Law School, University of Bologna, Italy, antonino.rotolo@unibo.it

Abstract. In this paper we propose a temporal extension of Defeasible Logic to model legal modifications, such as abrogation and annulment. Hence, this framework overcomes the difficulty, discussed elsewhere [7], of capturing these modification types using belief and base revision.

1 Introduction

The aim of our work is to study the notion of legal modification in Defeasible Logic (DL). Legal modifications are the ways through which the law implements norm dynamics. Typically, the law regulates its own changes and introduces norms whose peculiar objective is to change the system by specifying what and how other existing norms should be modified [5,6]. We are interested here in investigating the concepts of *abrogation* and *annulment*. *Annulment* is seen as a kind of repeal, as it makes a norm invalid and removes it from the legal system. Its peculiar effect applies *ex tunc*: annulled norms are prevented to produce all their legal effects, independently of when they are obtained. *Abrogation*, too, corresponds to a type of norm removal, even though it is different from annulment; the main point is that abrogations usually operate *ex nunc* and so do not *retroactively* cancel the effects that were obtained before the modification. *Retroactivity* is crucial. However, the distinction is not sometimes so sharp, as some cases of abrogation, too, admit that some (but not all) effects can retroactively be blocked [7,5,6]. Hence, in a nutshell, what we have to bear in mind here is that the law implements two different reasoning patterns in such a way as, in one case, norms are removed with *all* their effects, whereas in other cases norms are removed but *some or all* their effects propagate if obtained before the modification.

This paper is Part 2 of [7]. In [7] we argued that it is hard in DL to move to a general analysis, based on theory revision, where time is not considered. Rather, we proposed that dynamics of a legal system *LS* are correctly captured by a time-series $LS(t_1), LS(t_2), \ldots, LS(t_j)$ of its versions. Each version of *LS* is called a *norm repository*. The passage from one repository to another is effected by legal modifications or simply by persistence [6]. But dynamics of norm change and retroactivity need to introduce another time-line within each version of *LS*. Clearly, retroactivity does not imply that we can really change the past, but it rather requires that we have to reason on the legal system from the *viewpoint* of its current version but as it were revised in the past: when we change some $LS(i)$ retroactively, this does not mean that we modify some $LS(k)$,

3rd International Workshop on Normative Multiagent Systems (NorMAS 2008), Luxembourg, 15-16 July, 2008

$k < i$, but that we move back from the perspective of $LS(i)$. Hence, we can "travel" to the past along this inner time-line, i.e. from the viewpoint where we modify norms.

The layout of this Part 2 is as follows. Section 2 overviews the basics of DL. Section 3 proposes a temporal extension of DL able to correctly model abrogation and annulment: Section 3.1 describes the new formal language; Section 3.2 states the proof theory; Section 3.3 applies the framework to represent abrogation and annulment.

2 Defeasible Logic

Dynamics of legal change points out the importance of defeasibility due to the addition of new premises that can invalidate formerly derivable effects. This means that norm modifications proceed on the basis of defeasible reasoning [7]. In fact, the reasoning used in this context forms part of the wider domain of legal reasoning, which too is deemed to be defeasible [11]. In line with [5,6], we will show how to model defeasible reasoning on legal modifications using DL [10,1]. DL is based on a logic programming-like language [2] with an argumentation semantics [4]. In addition DL has linear complexity [9] and also has several efficient implementations [3].

A DL theory consists of a set of indisputable statements (facts), a set of rules, and a superiority relation over rules. We have three types of rules in DL: strict rules, defeasible rules and defeaters. A rule expresses a relationships between a set of premises and a conclusion, and the three types of rules convey the strength of the relationships. A strict rule states the strongest kind of relationship since the conclusion always holds when the premises are indisputable. Defeasible rules cover the case when the conclusion normally holds when the premises tentatively hold; finally defeaters consider a situation where the premises do not warrant the conclusions: in defeaters the premises simply prevent another rule to support the opposite. Accordingly, a conclusion can be labelled either as definite or defeasible. A definite conclusion is an indisputable conclusion, while a defeasible conclusion can be retracted if additional premises become available. DL is based on a constructive proof theory based for conclusions. Accordingly we can say that a derivation for a conclusion exists and that it is not possible to give a derivation for a conclusion. Based on these two ideas, we have four possibilities:

- $\pm\Delta p$, meaning that we (do not) have a definite proof for p (a definite proof is a proof where we use only facts and strict rules);
- $\pm\partial p$, meaning that we (do not) have a defeasible proof for p.

In what follows we will refer to $+\Delta$, $-\Delta$, $+\partial$ and $-\partial$ as proof tags, see [1] for formal conditions under which we can label a conclusion with one of these proof tags; in Section 3.2 we will give proof conditions for the temporal extensions we propose.

Strict (or definite) proofs are just derivations based on detachment for strict rules. Given a strict rule $a_1, \ldots, a_n \rightarrow b$, where we have definite proofs for all a_i's, we can deduce b. DL is a sceptical non-monotonic formalism: when conflict between two conclusions (i.e., one is the negation of the other) arises, DL refrains to take a decision and we deem both as not provable unless we have some more pieces of information that can be used to solve the conflict. One way to solve conflicts is to use a superiority relation over rules. The superiority relation gives us a preference over rules with conflicting conclusions. In case we have a conflict between two rules we prefer the conclusion of the strongest of the two rules. The superiority relation is applied in defeasible proofs. Defeasible proofs are structured in three phases: in the first phase we look for an argument supporting the conclusion we want to prove. More precisely we want an applicable rule for the conclusion. In the second phase we look for arguments/rules for the opposite of what we want to prove. In the last phase we rebut the counterarguments. This can be done by showing that the counterargument is not founded (i.e., some of the premises do not hold), or by defeating the counterargument, i.e., the counterargument is weaker than an argument for the conclusion we want to prove. In other words, a conclusion p is derivable when: (1) p is a fact; or (2) there is an applicable strict or defeasible rule for p, and either (2.1) all the rules for $\neg p$ are discarded (i.e., not applicable) or (2.2) every applicable rule for $\neg p$ is weaker than an applicable strict or defeasible rule for p. Note that a strict rule can be defeated only when its antecedent is defeasibly provable.

3 Temporal Defeasible Logic

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 [8]; in addition, it was suggested that the notion of a temporal viewpoint may solve the problem of retroactive modifications [5,6]. We present in this section some variants that deal with temporal dimensions as recalled above and presented in [7]. Dynamic aspects of legal reasoning are captured by two means: by first introducing temporal coordinates and, second, normative modifications.

[8] 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 besides the straightforward extension of the conditions given above, 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*.

However, the other components of our knowledge, too, have their temporal validity: 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 of 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, this applies as well to rules and then to derivations.

Let us consider the following example from a hypothetical taxation law. If the taxable income of a person at January 31, for the previous year is in excess on 100,000\$, then the top marginal rate computed at February 28 is 50% of the total taxable income.

And this provision is in force from January 1. This rule can be written as follows:

$$(Threshold^{31Jan} \Rightarrow HighMarginalRate^{28Feb})^{1Jan}$$

Let us suppose that the last instalment for the salary was paid to an employee on January 4, and that it makes the total taxable income greater than the threshold stated above. We used $Threshold^{4Jan}$ to signal that the threshold of 100,000$ has been certified on January 4. Clearly $Threshold^{4Jan}$ is a persistent property, thus in this case we can derive that the threshold was reached by January 31. So let us ask what the top marginal rate for the employee is if she lodges a tax return on January 20. What we have to do is to see whether the rule is still in force on January 20. Given that the norm was valid from January 1, and no changes were made to the legislation in between, the rule persists. Thus from the point of view of January 20, the top marginal rate is 50%. Suppose now that there is a change in the legislation and that the above norm is changed on February 15, and the change is that the top marginal rate is 30%.

$$(Threshold^{31Jan} \Rightarrow MediumMarginalRate^{28Feb})^{15Feb}$$

In this case if the employee lodges her tax return after February 15, the top marginal rate is 30% instead of 50%.

From the above example it is clear that 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 the above example illustrates 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 norm changes, i.e., to represent the different versions of a legal system, we use the notion of 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.

3.1 Language

The language of TDL is based on a (numerable) set of atomic proposition $Prop = \{p, q, \dots\}$, a set of rule labels $\{r_1, r_2, \dots\}$, a discrete totally ordered set of instants of time $\mathscr{T} = \{t_1, t_2, \dots\}$, the negation sign $\neg$, and the rule signs $\rightarrow$ (for strict rules), $\Rightarrow$ (for defeasible rules) and $\leadsto$ (for defeaters). 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., $t \in \mathscr{T}$, then l^t is a *temporalised literal*. If l^t is a temporalised literal and $x \in \{tran, pers\}$, then $l^{(t,x)}$ is a *duration literal*. If $l^{(t,x)}$ is a duration literal, $y \in \{tran, pers\}$ $t' \in \mathscr{T}$, then $l^{(t,x)}@(t', y)$ is a *fully temporalised literal*.

A *rule* is a relation between a set of of premises (conditions of applicability of the rule) and a conclusion. In this paper the admissible conclusions are either literals or rules themselves; in addition 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 system. In what follows we will use Rules to denote the set of rules, and MetaRules for the set of meta-rules, i.e., rules whose consequent is a rule.

A *temporalised rule* is either an expression $(r:\bot)^{(t,x)}$ (the void rule) or $(r:\emptyset)^{(t,x)}$ (the empty rule) or $(r:A \hookrightarrow B)^{(t,x)}$, where r is a rule label, A is a (possibly empty) set of temporalised literals, $\hookrightarrow$ is a rule sign, B is a duration literal, $t \in \mathscr{T}$ and $x \in \{tran, pers\}$.

We have to consider two temporal dimensions for norms in a normative system. The first dimension is when the norm is in force in it, and the second is when the norm exists in the normative system from a certain viewpoint. So far temporalised rules capture only one dimension, the time of force. To cover the other dimension we introduce the notion of temporalised rule with viewpoint. A *temporalised rule with viewpoint* is an expression $(r:A \hookrightarrow B)^{(t,x)} @ (t',y)$, where $(r:A \hookrightarrow B)^{(t,x)}$ is a temporalised rule, $t' \in \mathscr{T}$ and $y \in \{tran, pers\}$.

Finally, we introduce meta-rules, that is, rules where the conclusion is not a simple duration literal but a temporalised rule. Thus a *meta-rule* is an expression $(s:A \hookrightarrow (r: B \hookrightarrow C)^{(t',x)}) @ (t,y)$, where $(r:B \hookrightarrow C)^{(t',x)}$ is a temporalised rule, $r \neq s$, $t \in \mathscr{T}$ and $y \in \{tran, pers\}$. Notice that meta-rules carry only the viewpoint time (the validity time) but not the "in force" time. The intuition behind this is that meta-rules yield the conditions to modify a legal system. Thus they specify what rules (norms) are in a normative system, at what time the rules are valid, and the content of the rules. Accordingly, these rules must have an indication when they have been inserted in a normative system, but then they are universal (i.e., apply to all instants) within a particular instance of a normative system.

Every temporalised rule is identified by its rule label and its time. Formally we can express this relationship by establishing that every rule label r is a function $r : \mathscr{T} \mapsto$ Rules. Thus a temporalised rule r^t returns the value/content of the rule 'r' at time t. This construction allows us to uniquely identify rules by their labels[3], and to replace rules by their labels when rules occur inside other rules. In addition there is no risk that a rule includes its label in itself. In the same way a temporalised rule is a function from $\mathscr{T}$ to Rules, we will understand a temporalised rule with viewpoint as a function with the following signature: $\mathscr{T} \mapsto (\mathscr{T} \mapsto \text{Rules})$. A legal system LS is a sequence of versions $LS(t_0), LS(t_1), \ldots$. The temporal dimension of viewpoint corresponds to a version while the temporal dimension temporalising a rule corresponds to the time-line inside a version. 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 $LS(t_r)$. Accordingly, a version of LS is just a repository (set) of norms (implemented as temporal functions). Given a rule r, the expression $r^t @ t'$ gives the value of the rule (set of premises and conclusion of the rule) at time t in the repository t'. The content of a void rule, e.g., $(r:\bot)^t @ t'$ is $\bot$, while for the empty rule the values is the empty set. This means that the void rule has value for the combination of the temporal parameters, while for the empty rule, the content of the rules does not exist for the given temporal parameters.

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

[3] We do not need to impose that the function is an injection: while each label should have only one content at a given time, we may have that different labels (rules) have the same content.

of defeaters in R by R_{dft}. $R[q]$ denotes the set of rules in R with consequent q. For a rule $(r : A \hookrightarrow B)^{(t,x)}@(t',y)$ or a meta-rule $(r : A \hookrightarrow B)@(t,x)$ we will use $A(r)$ to indicate the body or antecedent of the rule, i.e., A, and $C(r)$ for the head or consequent of the rule, i.e., B. Given a temporalised rule $(r : A \hookrightarrow B)^{(t,x)}$,

$$R[\sim r^t] = \{(r : \bot)^{(t,x)}\} \cup \{(r : \emptyset)^{(t,x)}\} \cup \{(r : A' \hookrightarrow B')^{(t,x)} | A' \neq A \text{ or } B' \neq B\}$$

Finally, for every literal, rule, and every temporal dimension, we have the specification 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 $l^{(t,tran)}$, 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 $l^{(t,x)}@(t',tran)$ the reading is that the validity of l at t is specific to the legal system corresponding to repository associated to t', while $l^{(t,x)}@(t',pers)$ 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 $r^{(t,tran)}$ 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).

We will often identify rules with their labels, and, when unnecessary, we will drop the labels of rules inside meta-rules. Similarly, to simplify the presentation and when possible, we will only include the specification whether an element is persistent or transient only for the elements for which it is relevant for the discussion at hand.

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 system. Thus a temporalised rule r^t gives the 'content' of the rule 'r' at time t; in legal terms it tells us that norm r is in force at time t. The expression $(p^{t_p}, q^{t_q} \Rightarrow (p^{t_p} \Rightarrow s^{(t_s,pers)})^{(t_r,pers)})@(t,tran)$ means that, the repository at t, if p is true at time t_p and q at time t_q, then $p^{t_p} \Rightarrow s^{(t_s,pers)}$ is in force from time t_r onwards.

A legal system is represented by a temporalised defeasible theory, i.e. 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 (i.e., 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 $\mathscr{T} \mapsto (\mathscr{T} \mapsto \text{Rules} \times \text{Rules})$.

An unmodifiable rule is a rule such that $\forall t,t',t'',t'''\ r^t@t' = r^{t''}@t'''$. This means the content/value of the rule is the same across all repositories for all instants. The superiority relation $\prec$ determines the relative strength for rules for every instant in every version of the legal system. Thus it is possible that a rule r is both stronger and weaker than another rule s in two versions of the legal system, and then that two rules in different repositories have opposite relative strengths. To illustrate this case, consider water restrictions in force in South East Queensland in January 2007, where it is permitted to water garden in residential properties on Tuesday, Thursday and Saturday for odd number properties and on Wednesday, Friday and Sunday for even number properties;

and watering is otherwise forbidden. This regulation can be represented as follows

$$r: \Rightarrow \neg watering, \quad o: OddNumber \Rightarrow watering, \quad e: EvenNumber \Rightarrow watering$$

where the superiority contains, among others:

$$o \prec^{2007}_{Monday} r, \quad r \prec^{2007}_{Tuesdays} o, \quad e \prec^{2007}_{Monday} r, \quad r \prec^{2007}_{Wednesdays} e$$

Hence, in 2007, on Tuesday rule e is stronger than rule r, but on Monday r is stronger than e.

3.2 Proof Conditions

We are now ready to define how conclusions can be obtained in TDL. Notice that the main difference between the proof conditions given here and those of basic DL (of course besides the presence of the temporal dimensions) is that, in basic DL, rules are always given as elements of the theory, while here every time we have to use a rule, we have to ensure that the rule is derivable from the theory. Given the structure of a theory and the types of the rules we have, the 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.

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 of 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:

- $+\Delta t@t'\ x^{t_x}$: we have a definite derivation of x^{t_x} at time t using the elements in the repository at time t';
- $-\Delta t@t'\ x^{t_x}$: we can show that it is not possible to have a definite derivation of x^{t_x} at time t using the elements in the repository at time t';
- $+\partial t@t'\ x^{t_x}$: we have a defeasible derivation of x^{t_x} at time t using the elements in the repository at time t';
- $-\partial t@t'\ x^{t_x}$: we can show that it is not possible to have a definite derivation of x^{t_x} at time t using the elements in the repository at time t'.

In the presentation of the proof conditions we will adopt 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 repository, t_r is the repository time, thus it is the time-line of the legal system as a whole. Finally, the last temporal dimension is the object time, which in the case of 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 general mechanism for a derivation in the present framework is as follows. First of all, 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 legal 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 element 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 two previous steps, and we check whether 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) $\exists s @ t'_r \in R_s^{\text{meta}} : \forall a^{t_a} \in A(s), +\Delta t_d @ t_r\ a^{t_a} \in P[1..n]$, or
3) $+\Delta t'_d @ t''_r\ r^{t'_v}$.

where:

1. if r is persistent, then $t'_v \le t_v$; if r is transient, then $t_v = t'_v$;
2. if facts, rules and meta-rules are persistent across repositories, then $t'_r < t_r$, otherwise $t'_r = t_r$;
3. $t'_d < t_d$ if conclusions are persistent within a repository; otherwise $t'_d = t_d$;
4. $t''_r < t_r$ if conclusions are persistent across repositories; otherwise $t''_r = t_r$.

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., the 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 persistence or by a direct derivation with the appropriate time reference depends on the proof conditions for literals and the variant of TDL 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
2.1) $r^{t'_v} @ t'_r \in R^{\text{mod}}$ or $\exists s^{t_s} \in R_{sd}^{\text{meta}}[r^{t'_v}] : \forall a^{t_a} \in A(s), +\partial t'_d @ t''_r\ a^{t_a} \in P[1..n]$ 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 $r^{t'_v} @ t'_r \in R^{\mathrm{mod}}$ or $m^{t_m} \prec^{t_r}_{t_d} s^{t_s}$, if $r^{t'_v} @ t'_r \notin R^{\mathrm{mod}}$ or

.3) $\exists w^{t_w} \in R[r^{t''_v}] : \forall c^{t_c} \in A(w), +\partial t'''_d @ t''''_r\ c^{t_c} \in P[1..n]$ and $m^{t_m} \prec^{t_r}_{t_d} w^{t_w}$

where

1. if r is persistent, then $t'_v \leq t_v$; if r is transient, then $t_v = t'_v$;
2. 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. $t''_d \leq t_d$, $t'''_d \leq t_d$); 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. $t''_d = t_d$, $t'''_d = t_d$);
3. 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 $t''_r, t'''_r, t''''_r \leq t_r$; otherwise $t''_r, t'''_r, t''''_r = t_r$
4. if $r^{t'_v}$ and s (i.e., facts, rules, and meta-rules) are persistent with respect to repositories, then $t'_r \leq t_r$; otherwise $t'_r = t_r$.

A rule r is defeasibly provable at time t_d, given the information available in a repository t_r, if (1) the rule is strictly provable with the same parameters, or (2) we have definitely rejected that the content of the rule is different from what we want to prove, and then we have some justification to the claim. This means that (2.1) the rule is given in theory $r \in R^{mod}$. In this case, the rule can be given with a previous validity time (t'_v, $t'_v < t_v$) if that parameter is labelled as persistent. Similarly for the enactment time (or in-force time) t'_r. Notice that for a given rule, there are no constraints for the derivation time (t_d): given rules are understood as universally valid for that temporal dimension. For the second part of (2.1) we have that there is a meta-rule having r as its conclusion. In this case we have to check that the antecedent of the rule has been available to make the rule applicable at the derivation time t_d. This aspect depends on the particular variant of TDL one wants to adopt. The antecedent could have been derived at a previous time t_a, $t_a < t_d$, in a variant where conclusions persists within a repository; or with the same derivation time $t_a = t_d$, but in a previous version of the repository $t'_r < t_r$ if conclusions persist over repositories. We will fully explain these concepts when we present the proof conditions for literals. Clause (2.2) ensures that there are no justified reasons to claim the content of the rule different from what we want to prove. Remember the given interpretation of rules (rule labels) as function from the temporal dimensions to the content of a rule (i.e., the relationships between the antecedent, a set of premises and the conclusion). Thus for every combination of temporal parameters for a rule, there is only a single value for the content of the rule. Thus if we want to prove $+\partial 10@1\ (r : a^{t_a} \Rightarrow b^{t_b})$ we have to ensure that there is not way in which the content of rule r at time 10 in repository 1 is different from $a^{t_a} \Rightarrow b^{t_b}$.

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) $\exists r^{t_v} \in R_s[p^{t'_p}]$, $+\Delta t_d @ t_r\ r^{t'_v} \in P[1..n]$, $t'_v = t_d$ and
$\forall a^{t_a} \in A(r) : +\Delta t_d @ t_r\ a^{t_a} \in P[1..n]$; or

3) $+\Delta t'_d @ t'_r\ p^{t'_p} \in P[1..n]$.

where:

1. if p is persistent, then $t'_p \leq t_p$; if p is transient, then $t'_p = t_p$;
2. if r is persistent, then $t'_v \leq t_v$; 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. if conclusions are persistent within a repository, then $t'_d < t_d$; otherwise $t'_d = t_d$;
5. if conclusions are persistent across repositories, then $t'_r < t_r$; otherwise $t'_r = t_r$.

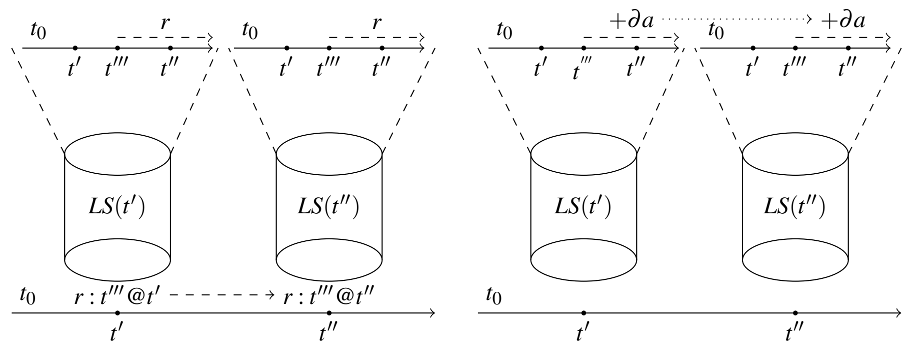
What we want to point out for strictly literal provability is the mechanism governing persistence of conclusions. While persistence of rules and facts (within or across repositories) is a property of the single instances, persistence of conclusions is a property characterising variants of TDL. In this case, a conclusion is persistent within a repositories if it is possible to carry over a derivation from one instant to a successive instant while keeping the time reference relative to the repository unchanged. This means that, for example, if one is able to prove $+\Delta t@1\ p$, for $t = 10$ then for all $t' > 10$, $+\Delta t'@1\ p$ can be proved. Notice that in this case all we have to do is to provide a strict proof for p at time t using the information in the repository at time 1. For persistence of conclusions across repositories, on the other hand, we keep fixed the derivation time, but once a conclusion has been proved in a repository, it can be used in all repositories succeeding it. Thus, for example, if we prove $+\Delta 10@t\ p$, with $t = 2$, then for all $t' > 2$, we have $+\Delta 10@t'\ p$. Notice that, in this case, it is possible, as it often happens with abrogation (see Section 3.3 for details), that the reason for proving p in repository t no longer subsists in repositories after t. The two types of persistence of conclusions can be combined.

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) $r^{t'_v}@t_r \neq \emptyset$, $r^{t'_v}@t'_r \in R_{sd}[p^{t'_p}]$, $+\partial t'_d@t'_r\ r^{t'_v} \in P[1..n]$ 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 $t'_d < t_d$), $t'_d \leq t''_d \leq t_d$; otherwise $t'_d = t''_d = t_d$;
5. if conclusions are persistent over repositories, $t'_r \leq t''_r \leq t_r$; otherwise $t'_r = t''_r = t_r$.



(a) Rule Persistence. A persistent rule r enacted at time t' and in force at t''' carries over from the legal system $LS(t')$ to the legal system $LS(t'')$, where it is still in force at t'''.

(b) Causal Conclusion Persistence. A conclusion is causal if it persists from $LS(t')$ to $LS(t'')$ even if the rules used to derive it are no longer effective in $LS(t'')$.

Fig. 1. Rules and Conclusions Persistence

The conditions to defeasibly derive literals, inherit intuitions from standard DL and the features described for the other types of derivation. The mechanism of persistence of conclusion over derivation (or within a repository) is essentially the same as that of strict conclusions. The main difference regards the way conclusions persist across repositories. In this case it is not enough that a defeasible derivation existed in previous repository. What is required is that the rule used to prove the conclusion still exists in the current repository w.r.t. the time it was valid in the previous repository, plus the conclusions that are to be proved can carry over from one repository to successive ones. Thus, for example, if we are able to prove $+\partial 10@1\ p^{t_p}$ because we can prove $+\partial 10@1\ (r : a^{t_a} \Rightarrow p^{t_p})$ and $+\partial 10@1\ a^{t_a}$, then we must be able to prove $+\partial 10@2\ a^{t_a}$ and that r has not been revoked after 1.

The proof conditions given above produce classes of variants of TDL, according to conditions on the temporal parameters. In particular, it is possible to define variants capturing different types of persistence. Of particular relevance to norm modifications we mention *rule persistence* and *causal conclusion persistence*. Generally once a norm has been introduced in a legal system, or better in a specific version of it, the norm continues to be in the system unless it is explicitly removed. This means that the norm must be included in all versions succeeding the one in which it has been first introduced (see Figure 1(a) for a graphical representation of this phenomenon). This effect is achieved by specifying that the derivation of rules is persistent over repositories. On the other hand, if we can prove a conclusion with respect to a specific version of the legal system in some cases we have to propagate it to successive versions. In particular, this is the case when we have causal conclusions. However, for some type of norm modifications, namely annulment, we have to block the persistence of conclusion over repositories when the reasons for deriving a conclusion are no longer in the system. See Figure 1(b) for a graphical representation of causal conclusion persistence. 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.

To illustrate these ideas consider the following theory:

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

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.[4] Given $a^{10}@1$ we can $+\partial 10@1\ b^{30}$, since b persists from 20 to 30. But the the second rule cannot be applicable, since its validity time is 15. Thus, to apply it we have to assume that derivations are persistent within a repository. If this is the case then we obtain $+\partial 15@1\ b^{30}$, which 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. The point to note for conclusion persistence is that, if we have a derivation in a preceding repository and the derivation is not 'killed' in successive repositories, we can carry over the conclusion from the repository where the conclusion has been proved to successive repositories.

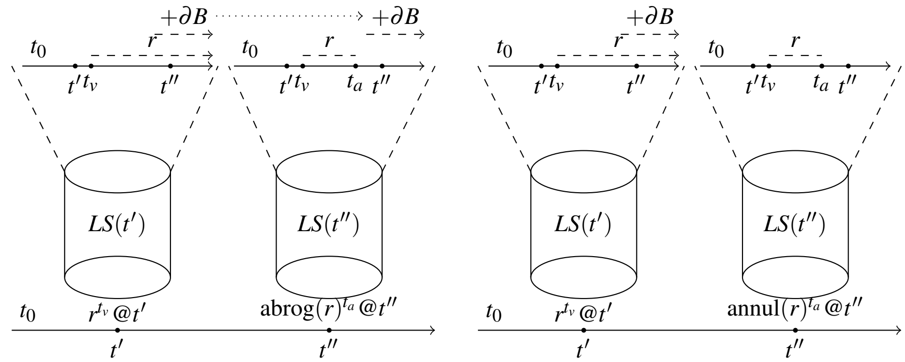
3.3 Abrogation and Annulment in TDL

Let us apply TDL to abrogation and annulment. Both modifications cancel norms from a legal system. Let $LS(t')$ be the repository containing a modifiable rule $(r : A \hookrightarrow B)^{(t_v,pers)}$ such that $t' < t_v < t''$, and $LS(t'')$ be the subsequent repository where we apply the modification of r, which is effective from a certain time t_a. Then $LS(t'')$ will contain $(r : \emptyset)^{(t_a,pers)}$. This makes the previous version of r inapplicable in $LS(t'')$ from t_a, and so, there, we no longer obtain B using r:[5] condition (2.1) for Defeasible Literal Provability states that $(r : A \hookrightarrow B)$ is applicable if it is provable with that content, but this does not hold after the modification (see condition (2.2) for Defeasible Rule Provability).

Does this solution solve all problems? Suppose the modification is retroactive, such that $t_v < t_m < t''$. This means that $LS(t'')$ contains an applicable meta-rule such as $(mr : A' \hookrightarrow (r : \emptyset)^{(t_a,pers)})@t''$. Note that the effect of mr is persistent to guarantee that r is null from t_a onwards. With some examples of *abrogation*, this measure may work, as we block the derivation of B, based on r in $LS(t'')$, from t_a onwards. Accordingly, if B was derived (with its own appropriate time) in $LS(t')$, it can carry over from $LS(t')$ to $LS(t'')$ (see Figure 2(a)). But this does not apply to annulments, for which B can carry over only through the inner time-line of $LS(t')$ In [6] we suggested that the solution is that annulment is obtained by blocking persistency of derivations across repositories. In other words, conclusions of the annulled rule are only derived in the repository in which the modification does not occur (see Figure 2(b)). However, no technical solution was offered. Our solution is as follows. Let the positive defeasible extension of a theory T be the set $E^{+\partial}(T) = \{p^{t_p}@t | T \vdash +\partial\#t'@t\,p^{t_p}\}$.

[4] To make the example simpler we have used two different scales for the derivation time and the repository time. Anyway in legal reasoning these will be on the same time scale.

[5] For simplicity, let's explicitly reason for the moment on the repository time and time of force only. For example, B will also have a temporal parameter and, if persistent, will hold from then onwards. Let's assume that B is persistent and its time is slightly after t_a.



(a) Abrogation. In $LS(t')$ rule r produces a persistent effect B. B carries over by persistence to $LS(t'')$ even if r is no longer in force.

(b) Annulment. In $LS(t')$ rule r is applied and produces a persistent effect B. Since r is annulled in $LS(t'')$, B must be undone as well.

Fig. 2. Abrogation and Annulment

Abrogation Given a rule $(r : A \hookrightarrow b^{t_b})^{t_r}@t$, the abrogation of r at t_a in repository $t'$is defined as follows:

$$T_r^{abr(t_a,t')} = \begin{cases} T & \text{if } r \notin E^{+\partial}(T) \\ (F, R', \prec) & \text{otherwise} \end{cases} \tag{1}$$

where $R' = R \cup \{(abr_r : \Rightarrow (r : \bot)^{(t_a,pers)})@(t', pers)\}$, where $t' > t$. The abrogation simply terminates the applicability of the rule. More precisely this operation sets the rule to the void rule. The rule is not removed from the system, but it has now a form where no longer can produce effects. This is in contrast to what we do for annulment where the rule to be annulled is set to the empty rule. This amounts to removing the rule from the repository. From the time of the annulment the rule has no longer any value.

Annulment The definition of a modification function for annulment depends on the underlying variants of TDL, in particular whether conclusions persist across repositories. In a variant where conclusions do not persist over repositories, the operation can be simply defined by the introduction of a meta-rule setting the rule to be annulled to $\emptyset$, with the time when the rule is annulled and the time when the meta-rule is inserted in the the legal system. Thus to annul at t_a the rule $r^{t_r}@t$ we introduce a meta-rule $(mr : \Rightarrow (r : \emptyset)^{(t_a,pers)})@(t', pers)$.

In a variant where conclusions persist over repositories we need some additional technical machinery: given a set of duration literals D, a set of temporalised literals T and a total discrete ordered $(\mathscr{T}, <)$, we define

$$D \cap_{(\mathscr{T},<)} T = \{l^t \in T | \exists l^{(t',x)} \in D : t' = t \text{ if } x = tran, \text{ and } t' \leq t' \text{ otherwise}\}$$

Given a duration literal $b^{(t,x)}$ and a theory T, we defined the *dependence set*, i.e., the set of literals (called *critical literals*) potentially depending on it, as follows:

$$Dep(b^{(t,x)}) = \{b^{(t,x)}\} \cup \{c^{(t_c,x_c)} | \exists r \in R : C(s) = c^{(t_c,x_c)} \wedge A(r) \cap_{(\mathscr{T},<)} Dep(b^{(t,x)}) \neq \emptyset\}$$

Then, if the annulment applies at t_a in repository t'

$$T^{annul(t_a,t')}_{(r:a_1,\ldots,a_n \hookrightarrow b^{(t_b,x)})^{t_r} @ t} = \begin{cases} T & \text{if } r \notin E^{+\partial}(T) \\ (F, R', \prec') & \text{otherwise} \end{cases} \tag{2}$$

where[6]

$$\begin{aligned} R' = R \cup \{ & (r : \emptyset)^{(t_a,pers)} @ (t', pers), \\ & (r^{\sim} : \rightsquigarrow \sim b^{(t_b,x)})^{(t_a,pers)} @ (t', tran), \\ & (r^{ann} : \Rightarrow ann(b)^{(t_b,x)})^{(t_a,pers)} @ (t', tran)\} \\ \cup \{ & (s^{rep} : A(s) - Dep(C(r)) \cup \{ann(a)^{t_a} | a \in A(s) \cap_{(\mathscr{T},<)} Dep(C(r))\} \hookrightarrow \\ & \qquad ann(C(s))^{(t_a,pers)})^{(t_a,tran)} @ (t', tran), \\ & (s^{ann} : ann(C(s))^{t_a} \rightsquigarrow \sim C(r))^{(t_a,pers)} @ (t', tran) | \\ & \quad \text{if } A(s) \cap_{(\mathscr{T},<)} Dep(C(r)) \neq \emptyset\} \\ \cup \{ & (s^{nan} : A(s) \Rightarrow \sim ann(C(r))^{(t_a,tran)})^{(t_a,tran)} @ (t', tran) | \\ & \quad \text{if } C(s) \in Dep(C(r)) \wedge A(r) \cap_{(\mathscr{T},<)} Dep(C(r)) = \emptyset\} \\ \prec' = \prec \cup & \{(t', t_a, r^{rep}, r) | r \in R\} \cup \{(t', t_a, r^{nan}, s^{ann}) | r, s \in R\} \end{aligned}$$

The idea behind this construction is to introduce new (auxiliary) literals to signal whether literals are eventually revoked (declared null) as a consequence of an annulment. Then, for every rule where literals depending on the conclusion of the rule to be annulled occur in the antecedent, we create a copy of the rule where all critical literals are replaced by auxiliary literals. Moreover, for each critical literal its auxiliary literal is the body of a defeater for the complement of the critical literal. Finally, for each rule for a critical literal different from the conclusion of the rule to be annulled where no critical literal appears in the antecedent, we create a defeasible rule with the same body and as conclusion the complement of the critical literal.

Note that the above construction guarantees that for every pair l, $ann(l)$ at most one of them is defeasibly provable, and that, if the strict part of the theory is consistent, then if $+\partial t@t'\ ann(l)$, then $-\partial t@t'\ l$. The intuition here is that the introduction or the meta-rule setting the rule to be annulled to $\bot$ determines that we no longer carry over the conclusion of the rule from one repository to the next one. However, this does not prevent conclusions depending on it to pass over (after all at the time they were derived we had then valid reasons to derive them, and unless some preventing reasons occurred after, we have no reasons to stop them to pass from one repository to next

[6] To simplify the notation in the rest of the definition the rules are the conclusion of a meta-rule (each with a unique name), thus the expression $(r : a_1, \ldots, a_n \hookrightarrow b^{t_b})^{t_r} @ t$ must be understood as the abbreviation of the meta-rule (with empty body) $(mr : \Rightarrow (r : a_1, \ldots, a_n \hookrightarrow b^{t_b})^{t_r}) @ t$.

one). Hence, we need specific reasons to stop them. Thus the idea to refute them is to add the explanation that they were derived from 'causes' declared null in a successive step, and thus they must be null as well.

Example 1. Consider a legal system at time 1 encoded in the following theory T:

$$F = \{a^{10}@(1,pers), c^{10}@(1,pers), f^{10}@(1,pers)\},$$
$$R = \{(r_1 : a^{10} \Rightarrow b^{(10,pers)})^{(10,pers)}@(1,pers),\ (r_2 : b^{10}, c^{10} \Rightarrow d^{(10,pers)})^{(10,pers)}@(1,pers),$$
$$(r_3 : d^{10} \Rightarrow e^{(10,pers)})^{(10,pers)}@(1,pers),\ (r_4 : f^{10} \Rightarrow e^{(10,tran)})^{(10,pers)}@(1,pers)\}$$
$$\prec = \emptyset.$$

Clearly we can prove $+\partial 10@1\ X$ for $X \in \{a,b,c,d,e,f\}$. Now suppose that the legal system is changed by revoking rule r_1, and that the change is valid from 10. The resulting legal system is the legal system at time 2. If the change is an abrogation, then, the resulting legal system at time 2 is obtained by the addition of the rule $(r_1^{abr} : \bot)^{(10,pers)}@(2,pers)$. The legal system at 2 is obtained by adding the rules (meta-rules) implementing the annulment function $T_{r_1}^{ann(10,2)}$. Namely we revise T by introducing the rules

$$(r_1 : \emptyset)^{(10,pers)}@(2,pers),\ \ (r_1^{\sim} : \rightsquigarrow b^{(10,pers)})^{(10,pers)}@(2,tran),$$
$$(r_1^{ann} : \Rightarrow ann(b)^{(10,pers)})^{(10,pers)}@(2,tran),$$
$$(r_2^{rep} : c^{10}, ann(b)^{10} \Rightarrow ann(d)^{(10,pers)})^{(10,pers)}@(2,tran),$$
$$(r_2^{ann} : ann(d)^{10} \rightsquigarrow \neg d^{(10,pers)})^{(10,tran)}@(2,tran),$$
$$(r_3^{rep} : ann(d)^{10} \Rightarrow ann(e)^{(10,pers)})^{(10,pers)}@(2,tran),$$
$$(r_3^{ann} : ann(e)^{10} \rightsquigarrow \neg e^{(10,pers)})^{(10,tran)}@(2,tran),$$
$$(r_4^{nan} : f^{10} \Rightarrow \neg ann(e)^{(10,tran)})^{(10,tran)}@(2,tran).$$

From the point of view of the legal system at 1 we have a derivation of b^{10} at 10 ($+\partial 10@1\ b^{10}$). Thus, allowing conclusions to persist over repositories, would mean that we can carry over the derivation of it to the repository at 2 (Clause 2.1 of Defeasible Literal Provability, plus condition on conclusion persistence). But setting rule r_1 to $\emptyset$ in 2 produces the effect that now the rule no longer exists and thus it cannot longer be used. Hence we block the derivation b^{10}, more precisely, $-\partial 10@2\ b^{10}$. When we look at d, without rules r_2^{rep} and r_2^{ann}, d^{10} ($+\partial 10@1\ D^{10}$) was obtained from the viewpoint of 1. Rule r_2 has not been revoked, and at the time the conclusion was derived, the rule was applicable (i.e., the antecedent was provable). Thus, the conclusion passes from 1 to 2, that is $+\partial 10@2\ d^{10}$ would be derivable. However, this conclusion was the result of an act declared null by the (retroactive) annulment. Finally, for e we have that there are two rules for it. In the first rule (r_3) where e depends on some annulled literal, but this is not the case for the second rule (r_4). In the annulment r_3 generates r_3^{rep}, and r_4 generates r_4^{nan}. These two rules are in conflict which each other, but $r_3^{rep} \prec_{10}^{2} r_4^{nan}$, thus r_4^{ann} prevails, and we are able to prove $ann(e)$, $-\partial 10@2\ ann(e)$, so r_3^{ann} is not applicable at 10 w.r.t. repository 2. Hence we can use r_4 to continue to derive $+\partial 10@2\ e$.

4 Summary

In this paper we extended the logic presented in [5] to capture different temporal aspects of abrogations and annulments. 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 to deal with the derivation of rules and literals. In particular, for each proof condition we identified several temporal constraints that permit to allow for, or block, persistency with respect to specific time-lines. This virtually leads to define different variants of TDL according to whether a condition is adopted or not. Then we described some issues related to norm modifications and we illustrated the techniques with respect to annulment and abrogation. We showed that the temporal formalism introduced here is able to deal with complex scenarios such as retroactivity. 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.

Acknowledgements National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program. Antonino Rotolo was supported by the EU project ESTRELLA (IST-2004-027655).

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. G. Antoniou, D. Billington, G. Governatori, and M.J. Maher. Embedding defeasible logic into logic programming. *Theory and Practice of Logic Programming*, 6(6):703–735, 2006.
3. N. Bassiliades, G. Antoniou, and I. Vlahavas. DR-DEVICE: A defeasible logic system for the Semantic Web. In *Proc. PPSWR03*, 2004.
4. 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.
5. G. Governatori, M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Norm modifications in defeasible logic. In *Proc. JURIX05*. IOS Press, 2005.
6. G. Governatori, M. Palmirani, R. Riveret, A. Rotolo, and G. Sartor. Variants of temporal defeasible logic for modelling norm modifications. In *Proc. ICAIL07*. ACM Press, 2007.
7. G. Governatori and A. Rotolo. Changing legal systems: Abrogation and annulment. Part I: Revision of defeasible theories. To appear in *DEON* 2008.
8. G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *Proc. ICAIL05*, pages 25–34. ACM Press, 2005.
9. M.J. Maher. Propositional defeasible logic has linear complexity. *Theory and Practice of Logic Programming*, (6):691–711, 2001.
10. D. Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press, 1993.
11. G. Sartor. *Legal Reasoning*. Springer, Dordrecht, 2005.