Characterising Deadlines in Temporal Modal Defeasible Logic

Governatori, Guido; Hulstijn, Joris; Riveret, Régis; Rotolo, Antonino

published in
AI 2007: Advances in Artificial Intelligence
2007

document version
Publisher's PDF, also known as Version of record

Characterising Deadlines in Temporal Modal Defeasible Logic

Guido Governatori[1], Joris Hulstijn[2], Régis Riveret[3], and Antonino Rotolo[3]

[1] School of ITEE, The University of Queensland, Australia
guido@itee.uq.edu.au
[2] Vrije Universiteit, Amsterdam, The Netherlands
jhulstijn@feweb.vu.nl
[3] CIRSFID, University of Bologna, Italy
antonino.rotolo@unibo.it, regis.riveret@unibo.it

Abstract. We provide a conceptual analysis of several kinds of deadlines, represented in Temporal Modal Defeasible Logic. The paper presents a typology of deadlines, based on the following parameters: deontic operator, maintenance or achievement, presence or absence of sanctions, and persistence after the deadline. The deadline types are illustrated by a set of examples.

1 Introduction

Many normative rules in applications of artificial intelligence are concerned with deadlines. Bills must be paid before the end of the month; books need to be returned before a due date; products must be delivered within ten working days. In general we can say that a deadline combines a specification of a time point or condition δ, with the obligation to achieve some state of affairs φ *before* condition δ arrives.

Because deadlines combine time and obligations, they are naturally studied by a combination of temporal logic with deontic logic. Various papers use a combination of branching time temporal logic (CTL) with Standard Deontic Logic [5,4,3]. In such logics, the meaning of a deadline clause, i.e., that φ should occur before δ, can be expressed as an obligation which involves the 'until'-operator: $\mathrm{OBL}(\varphi \mathrm{U} \delta)$. This means that situations in which δ becomes true while φ has not been achieved, are considered to be a violation according to some normative system.

However, such a characterisation of deadlines does not indicate what will happen *after* the deadline. Is it required to still complete φ even after the deadline, as for late payments? Or is it impossible or unnecessary to complete φ after the deadline, as for the late delivery of a wedding cake? Is an explicit sanction enforced after the deadline, or is the sanction simply that achieving one's goal has become impossible?

Since deadlines can have different functions, it is likely that several notions of deadline can be distinguished. In this paper we therefore want to provide a conceptual analysis of various kinds of deadlines, and to provide a formal characterisation. Such a characterisation should make it easier to have intelligent systems reason about systems of deadline clauses, to find out for example whether they are mutually consistent, or how a course of action can be planned that meets all deadline clauses; compare the motivation given by [5].

M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 486–496, 2007.

To express the characterisation, we choose a Temporal Modal Defeasible Logic which combines the deontic modalities of obligation and permission with temporal intervals. Many normative rules allow for exceptions. Without defeasibility, it would be impossible to distinguish exceptions from violations. Moreover, a defeasible logic allows for a modular representation of deadline clauses, in which a clause may for example override earlier clauses, or override clauses issued by a lesser authority.

Usually, logic papers focus on the logical properties of a newly developed logic. In this paper, however, we would like to stress the conceptual issues of modelling temporal normative rules. It is necessary to understand deadlines, for example, for the implementation of procurement procedures, or for the management of automated service level agreements [10]. This research is part of a larger research effort to investigate the use of defeasible logic for reasoning with legal information [9,8].

The remainder of the paper is structured as follows. In Section 2 we define the Temporal Modal Defeasible Logic to be used in our characterisation. In Section 3 we provide a characterisation of different types of deadlines, and provide template formulas in the logic, to represent these types.

2 Temporal Modal Defeasible Logic

The logical framework of this paper is an interval based variant of Temporal Modal Defeasible Logic (TMDL), which is an extension of Defeasible Logic [2] to capture the deontic modalities obligation and permission and some aspects of time. Other variants of TMDL have proved useful in modelling normative reasoning [9,8] and cognitive agents [7].

We assume a linear discrete bounded set $\mathscr{T}$ of points of time termed 'instants' and over it the temporal order relation $>\subseteq \mathscr{T} \times \mathscr{T}$. We define intervals $[t_1,t_2]$ as sets of instants between the instants t_1 and t_2. Our framework is meant to incorporate temporal parameters into a non-monotonic model for deontic reasoning. It is not our intention to provide a general approach to temporal reasoning, nor an extensive analysis of operations and relations over time intervals, as e.g. [1].

A TMDL theory consists of a linear discrete bounded set of instants, a set of *facts* or indisputable statements, a set of *rules*, and a *superiority relation* $\succ$ among rules specifying when a rule may override the conclusion of another rule. A *strict rule* is an expression of the form $\varphi_1,\ldots,\varphi_n \rightarrow \psi$ such that whenever the premises are indisputable so is the conclusion. A *defeasible rule* is an expression of the form $\varphi_1,\ldots,\varphi_n \Rightarrow \psi$ whose conclusion can be defeated by contrary evidence. An expression $\varphi_1,\ldots,\varphi_n \rightsquigarrow \psi$ is a *defeater* used to defeat some defeasible rules by producing evidence to the contrary.

Definition 1 (Language). Let $\mathscr{T}$ be a linear discrete bounded ordered set of instants of time, in which the minimal unit is u and the lower and higher boundaries of $\mathscr{T}$ are denoted respectively by *min* and *max*. Let Prop be a set of propositional atoms, let Mod be a set of modal operators $\{\mathrm{OBL},\mathrm{PERM}\}$, and let Lab be a set of labels of rules. The sets below are defined as the smallest sets closed under the following rules:

Literals $\mathrm{Lit} = \mathrm{Prop} \cup \{\neg p | p \in \mathrm{Prop}\}$
Modal Literals $\mathrm{ModLit} = \{X\,l\,, \neg X\,l | X \in \mathrm{Mod}, l \in \mathrm{Lit}\}$

Intervals $\text{Inter} = \{T = [t_1,t_2] | [t_1,t_2] = \{x | t_1 \leq x \leq t_2\}, t_1,t_2 \in \mathscr{T}\}$
Temporal Literals $\text{TLit} = \{l{:}T | l \in \text{Lit},\ T \in \text{Inter}\}$
Temporal Modal Literals $\text{TModLit} = \{\varphi{:}T | \varphi \in \text{ModLit},\ T \in \text{Inter}\}$
Rules $\text{Rule}_s = \{(r \quad \varphi_1,\ldots,\varphi_n \rightarrow \psi) | r \in \text{Lab},\ \varphi_1,\ldots,\varphi_n,\psi \in \text{TLit} \cup \text{TModLit}\}$
$\text{Rule}_d = \{(r \quad \varphi_1,\ldots,\varphi_n \Rightarrow \psi) | r \in \text{Lab},\ \varphi_1,\ldots,\varphi_n,\psi \in \text{TLit} \cup \text{TModLit}\}$
$\text{Rule}_{dft} = \{(r \quad \varphi_1,\ldots,\varphi_n \rightsquigarrow \psi) | r \in \text{Lab},\ \varphi_1,\ldots,\varphi_n,\psi \in \text{TLit} \cup \text{TModLit}\}$
$\text{Rule} = \{\text{Rule}_s \cup \text{Rule}_d \cup \text{Rule}_{dft}\}$
Superiority Relations $\text{Sup} = \{s \succ r | s,r \in \text{Lab}\}$

We shall use some abbreviations: $A(r)$ denotes the set $\{\varphi_1,\ldots,\varphi_n\}$ of *antecedents* of the rule r, and $C(r)$ to denote the *consequent* ψ of the rule r. We also use $\text{Rule}[\varphi]$ for those rules in Rule whose consequent is φ. If γ is a literal, $\sim\gamma$ denotes the complementary literal: if q is the positive literal p then $\sim q$ is $\neg p$; and if q is $\neg p$, then $\sim q$ is p.

The resulting formalism permits to have expressions such as $\psi{:}[0,20]$ and $\text{OBL}\varphi{:}[1,10]$: the former says that ψ holds from 0 to 20, the latter that the obligation that φ applies from 1 to 10. Note that we allow for 'punctual intervals', i.e. intervals of the form $[t,t]$. As a notational convention, we simply write $\varphi{:}t$ instead of $\varphi{:}[t,t]$.

Definition 2 (TMDL Theory). A TMDL theory is a structure $D = (\mathscr{T}, F, R, \succ)$ where $\mathscr{T}$ a linear discrete bounded totally ordered set of instants of time, $F \subseteq \text{TLit} \cup \text{TModLit}$ is a finite set of facts, $R \subseteq \text{Rule}$ is a finite set of rules such that each rule has a unique label, $\succ \subseteq \text{Sup}$ is a set of acyclic superiority relations.

A conclusion of a theory D is a tagged literal φ having one of the following forms:

- $+\Delta\varphi$ and $-\Delta\varphi$, meaning respectively that φ *is*, or is *not* definitely provable in D,
- $+\partial\varphi$ and $-\partial\varphi$, meaning respectively that φ *is*, or is *not* defeasibly provable in D.

Provability is based on the concept of a derivation (or proof) in D. A derivation is a finite sequence $P = (P(1),\ldots,P(n))$ of tagged literals. Each tagged literal satisfies some proof conditions, which correspond to inference rules for the four kinds of conclusions we have mentioned above. Before moving to the conditions governing derivability of conclusions, we need to introduce some preliminary notions.

Definition 3 (Rule applicability). Let $\# \in \{\Delta, \partial\}$. If rule r is #-applicable then $\forall\varphi \in A(r)$, $+\#\varphi \in P(1..i)$. If r is #-discarded then $\exists\varphi \in A(r)$ such that $-\#\varphi \in P(1..i)$.

Definition 4 (Modality relations). For any $\varphi \in \text{ModLit}$, let the set $\text{convert}(\varphi)$ be:
$\text{convert}(\text{OBL}\varphi) = \{\text{PERM}\varphi, \neg\text{OBL}\sim\varphi\}$, $\text{convert}(\neg\text{OBL}\sim\varphi) = \{\text{PERM}\varphi\}$,
$\text{convert}(\text{PERM}\varphi) = \{\neg\text{OBL}\sim\varphi\}$, $\text{convert}(\neg\text{PERM}\sim\varphi) = \{\text{OBL}\varphi, \text{PERM}\varphi\}$.

Definition 5 (Intervals: basic notions). Let 'start()' and 'end()' be the functions that return respectively the lower and upper bounds of an interval. For any $T_1, T_2, T_3 \in \text{Inter}$,

- $T_1 \sqsubseteq T_2$ iff $\text{start}(T_2) \leq \text{start}(T_1)$ and $\text{end}(T_1) \leq \text{end}(T_2)$;
- $\text{over}(T_1, T_2)$ iff $\text{start}(T_1) \leq \text{end}(T_2)$ and $\text{start}(T_2) \leq \text{end}(T_1)$;
- $T_1 \sqcup T_2 = T_3$ iff $\text{end}(T_1) + u = \text{start}(T_2)$, $\text{start}(T_3) = \text{start}(T_1)$ and $\text{end}(T_3) = \text{end}(T_2)$.

We provide below the proof conditions to determine whether a temporal (modal) literal $\gamma{:}T$ is a definite conclusion of a theory D.

If $P(i+1) = +\Delta\gamma{:}T$ then
(1) $\gamma{:}T' \in F$ s.t. $T \sqsubseteq T'$, or
(2) $+\Delta\beta{:}T \in P(1..i)$ s.t. $\gamma \in \text{convert}(\beta)$, or
(3) $\exists r \in R_s[\gamma{:}T']$,
s.t. $T \sqsubseteq T'$, r is Δ-applicable, or
(4) $+\Delta\gamma{:}T' \in P(1..i)$ and $+\Delta\gamma{:}T'' \in P(1..i)$,
s.t. $T' \sqcup T'' = T$.

If $P(i+1) = -\Delta\gamma{:}T$ then
(1) $\gamma{:}T' \notin F$ s.t. $T \sqsubseteq T'$, and
(2) $-\Delta\beta{:}T \in P(1..i)$ s.t. $\gamma \in \text{convert}(\beta)$, and
(3) $\forall r \in R_s[\gamma{:}T']$,
s.t. $T \sqsubseteq T'$, r is Δ-discarded, and
(4) $-\Delta\gamma{:}T' \in P(1..i)$ or $-\Delta\gamma{:}T'' \in P(1..i)$,
s.t. $T' \sqcup T'' = T$.

Defeasible provability deals with conflicts. This requires to state when two literals are in conflict with each other:

Definition 6 (Complementary modal literals). Let $l \in \text{Lit}$ and $X \in \text{Mod}$. The set $\mathscr{C}(Xl)$ of complements of Xl is defined as follows:
$\mathscr{C}(\text{OBL } l) = \{\neg\text{OBL } l, \text{OBL}{\sim}l, \neg\text{PERM}l, \text{PERM}{\sim}l\}$ $\mathscr{C}(\neg\text{OBL}l) = \{\text{OBL}l, \neg\text{PERM}l\}$
$\mathscr{C}(\text{PERM}l) = \{\text{OBL}{\sim}l, \neg\text{PERM}l\}$ $\mathscr{C}(\neg\text{PERM}l) = \{\neg\text{OBL}{\sim}l, \text{PERM}l, \neg\text{PERM}{\sim}l\}$

As usual with defeasible logic, the derivation of a conclusion follows a three phase protocol. First we provide an argument in favour of the conclusion we want to prove. Second, we have to consider all possible counterarguments. Third, we have to rebut each of those counterarguments, i.e., find counterarguments against them.

In TMDL we must consider the times and intervals associated with the conclusion and the premises. So to prove a conclusion at time t we must find a rule of which the interval associated with the conclusion covers t. For counterarguments we look for rules producing the complement of the conclusion to be proved, and whose interval overlaps with the interval of the conclusion. The proof conditions are as follows.

If $P(i+1) = +\partial\gamma{:}T$ then
(1) $+\Delta\gamma{:}T \in P(1..i)$, or
(2) $-\Delta\beta{:}T' \in P(1..i)$,
s.t. $\beta \in \mathscr{C}(\gamma)$, over$(T', T)$, and
(2.1) $+\partial\beta{:}T' \in P(1..i)$, $\gamma \in \text{convert}(\beta)$, or
(2.2) $\exists r \in R_{sd}[\gamma{:}T']$,
s.t. $T \sqsubseteq T'$, r is ∂-applicable, and
(2.2.1) $\forall s \in R[\alpha{:}T']$,
s.t. $\alpha \in \mathscr{C}(\gamma)$, over$(T', T)$
(2.2.1.1) s is ∂-discarded, or
(2.2.1.2) $\exists w \in R[\gamma{:}T'']$, s.t. $T \sqsubseteq T''$,
w is ∂-applicable, and $w \succ s$, or
(3) $+\partial\gamma{:}T' \in P(1..i)$ and $+\partial\gamma{:}T'' \in P(1..i)$,
s.t. $T' \sqcup T'' = T$.

If $P(i+1) = -\partial\gamma{:}T$ then
(1) $-\Delta\gamma{:}T \in P(1..i)$, and
(2) $+\Delta\beta{:}T' \in P(1..i)$,
s.t. $\beta \in \mathscr{C}(\gamma)$, over$(T', T)$, or
(2.1) $-\partial\beta{:}T' \in P(1..i)$, $\gamma \in \text{convert}(\beta)$, and
(2.2) $\forall r \in R_{sd}[\gamma{:}T']$,
s.t. $T \sqsubseteq T'$, r is ∂-discarded, or
(2.2.1) $\exists s \in R[\alpha{:}T']$,
s.t. $\alpha \in \mathscr{C}(\gamma)$, over$(T', T)$
(2.2.1.1) s is ∂-applicable, and
(2.2.1.2) $\forall w \in R[\gamma{:}T'']$, s.t. $T \sqsubseteq T''$,
w is ∂-discarded, or $w \not\succ s$, and
(3) $-\partial\gamma{:}T' \in P(1..i)$ or $-\partial\gamma{:}T'' \in P(1..i)$,
s.t.$T' \sqcup T'' = T$.

Let us explain the proof condition of $+\partial\gamma{:}T$, the defeasible provability of $\gamma{:}T$. There are two cases: (i) show that $\gamma{:}T$ is already definitely provable (see (1)), or (ii) use the defeasible part of D. To prove $\gamma{:}T$ defeasibly we must show that its complements β are not definitely provable (see (2)). In clause (2.1), we show the provability of $\beta{:}T'$ such that β can be converted into γ (see Definition 4). In clause (2.2), we require that there

is a strict or defeasible rule $r \in R$ applicable, with $\gamma{:}T'$ as its head such that interval T' includes T. Now we consider possible attacks: reasoning chains in support of a complement α of γ: any rule $s \in R$ with a complement $\alpha{:}T'$ as its head and over(T', T). These attacking rules must be discarded (2.2.1.1), or must be counterattacked by a stronger rule w (2.2.1.2). Finally, in clause (3) we deal with the case where γ is defeasible provable on T' and T'', which form T.

The proof conditions for $-\partial\gamma{:}T$, that $\gamma{:}T$ is *not* defeasibly provable, are structured in a similar way. Positive derivations are replaced by negative ones and vice versa.

To illustrate the derivations, let us consider the following example:

$r_1\ b{:}5 \Rightarrow \text{OBL}a{:}[10,15]$ $\quad$ $r_2\ c{:}6 \Rightarrow \neg\text{OBL}a{:}[6,9]$
$r_3\ d{:}[5,10] \Rightarrow \neg\text{OBL}a{:}[12,20]$ $\quad$ $r_4\ e{:}[7,8] \Rightarrow \neg\text{OBL}a{:}[10,15]$

Given facts $\{b{:}[1,5], c{:}6, d{:}5,10, e{:}7\}$, we want to know if there is an obligation OBLa at time 10, i.e., $+\partial\text{OBL}a{:}10$. Since b holds from 1 to 5, rule r_1 is applicable, and we have an argument to support the conclusion. Then we have to consider the putative arguments against the obligation at the chosen time. The range of effectiveness of r_2 is $[6,9]$, so that does not overlap with the conclusion of r_1, namely $[10,15]$. The range of effectiveness of r_3 is $[12,20]$ which overlaps with the conclusion of r_1, but it does not cover the intended conclusion (10). So, we can discard this rule as well. Finally, the range of r_4 overlaps with the conclusion of r_1, and covers the time for the conclusion we want to derive. However, in this case, we can rebut the argument since its premise is that e holds from 7 to 8, but we are given that e holds only at 7, so we cannot use this rule. Since there are no more arguments against OBL$a{:}10$ we can conclude it.

3 Analysing Deadlines

In this section we provide a number of parameters to analyse deadline situations. The hypothesis is that different types of deadlines can be characterised by instantiations of these parameters. Each parameter corresponds to one or more template rules. By making combinations of different template rules, we can generate every possible type of deadline clause.

(1) Customers must pay within 30 days, after receiving the invoice.

Example (1) can be represented as follows. The deadline refers to an obligation triggered by receipt of the invoice (rule inv_{init}). After that the customer is obliged to pay. The obligation terminates when it is complied with (rule inv_{term}). The termination is modelled by a defeater rule ($\rightsquigarrow$). Note that the obligation itself may even persist after the deadline. Generally, a deadline signals that a violation of the obligation has occurred (rule inv_{viol}). This may or may not trigger an explicit sanction (see below).

inv_{init} $\text{get\_invoice}{:}t_1 \Rightarrow \text{OBLpay}{:}[t_1, max]$
inv_{term} $\text{OBLpay}{:}t_2, \text{pay}{:}t_2 \rightsquigarrow \neg\text{OBLpay}{:}t_2 + 1$
inv_{viol} $\text{get\_invoice}{:}t_1, \text{OBLpay}{:}t_1 + 31 \Rightarrow \text{viol(inv)}{:}t_1 + 31$

Suppose that the set of facts is $\{\text{get\_invoice}{:}0, \text{pay}{:}20\}$. We can derive $+\partial\text{get\_invoice}{:}0$, which makes rule inv_{init} applicable, leading to $+\partial\text{OBLpay}{:}[0, max]$, that is, an

obligation to pay applies indefinitely. Rule inv_{term} terminates the obligation at 21. Therefore rule inv_{viol} is not applicable, and we cannot derive a violation: $-\partial\text{viol(inv)}{:}30$.

This example can be turned into a general template for the representation of deadlines, where $\chi_1,..\chi_n$ are one or more contextual constraints, which trigger the existence of the obligation and fix time variables like t, φ is the condition to be achieved, and δ is the deadline condition. In the example above δ only specifies the 30 days period, but in general δ could be any action or event, such as 'having dinner', in 'I must finish work before dinner', when you do not exactly know when dinner will be. The violation fact $\text{viol}(r)$ is a specific literal, indexed by the name of the group of rules.

r_{init} $\chi_1{:}t_1,...,\chi{:}t_n \Rightarrow \text{OBL}\varphi{:}[t,max]$ (initialise)
r_{term} $\text{OBL}\varphi{:}t,\varphi{:}t \rightsquigarrow \neg\text{OBL}\varphi{:}t+1$ (terminate)
r_{viol} $\delta{:}t_\delta,\text{OBL}\varphi{:}t_\delta \Rightarrow \text{viol}(r){:}t_\delta$ (violation)

Achievement or Maintenance? Based on similar ideas about intentions and goals, we can distinguish *achievement obligations*, like example (1), from so called *maintenance obligations*, like example (2). For an achievement obligation, the condition φ must occur at least once before the deadline. For maintenance obligations, condition φ must obtain during all instants before the deadline. Here, the deadline only signals that the obligation is terminated. A violation occurs, when the obliged state does not obtain at some point before the deadline. Note that prohibitions, i.e. obligations to avoid some 'bad state', form a large class of maintenance obligations.

(2) Customers must keep a positive balance, for 30 days after opening an bank account.

pos_{init} open_account$:t_1 \Rightarrow \text{OBL}$positive$:[t_1,t_1+30]$
pos_{viol} OBLpositive$:t_2,\neg$positive$:t_2 \Rightarrow \text{viol(pos)}{:}t_2$

The generic formalisation for maintenance obligations consists of the following template formulas. Note that no termination rule is needed.

r_{init_main} $\chi{:}t,\delta{:}t_\delta \Rightarrow \text{OBL}\varphi{:}[t,t_\delta]$ (initialise maintenance)
r_{viol_main} $\text{OBL}\varphi{:}t,\neg\varphi{:}t \Rightarrow \text{viol}(r){:}t$ (violation of maintenance)

Persistence after the deadline. By definition, maintenance obligations do not persist after the deadline. But achievement obligations often do persist after the deadline, until they are achieved. However, this is not the case for all achievement obligations.

(3) A wedding cake must be delivered, before the wedding party.

In example (3), the obligation to deliver the cake does not persist after the deadline, since the wedding guests will have no use for it. In addition, the couple who ordered the cake, are entitled not to pay for the cake, or even claim damages after the deadline has passed without delivery. This can be seen as a kind of sanction.

In general, the difference between persistent and non-persistent achievement obligations will depend on conventions about the underlying goal of the deadline. Non-persistent obligations have a termination rule, similar to maintenance obligations.

wed_{init} order:t_1, wedding:$t_2 \Rightarrow$ OBLcake:$[t_1, t_2]$
wed_{term} OBLcake:t_3, cake:$t_3 \leadsto \neg$OBLcake:$t_3 + 1$
wed_{viol} wedding:t_2, OBLcake:$t_2 \Rightarrow$ viol(wed):t_2

The rules for non-persistent deadlines can be generalised as follows [1].

$r_{init\_non\_pers}$ χ:t, δ:$t_\delta \Rightarrow$ OBLφ:$[t, t_\delta]$ (initialise non-persistence)

Explicit or implicit Sanctions. Many normative rules are associated with an explicit sanction, such as payment of a fine. But this is not always the case. Often there is only an *implicit sanction*, of not being able to achieve the goal that underlies the deadline. For a sanction to be effective, it needs to be strongly disliked by the person sanctioned.

(4) Customers must pay within 30 days, after receiving the invoice. Otherwise, an additional fine must be paid.
(5) Customers must keep a positive balance for 30 days after opening an account, otherwise their account is blocked.
(6) A wedding cake must be delivered, before the wedding party. If not, the contract is breached and (for example) the customer may not have to pay.

An explicit sanction is often implemented through a separate obligation, which is triggered by a detected violation. In this setting, legislators may need further deadlines to enforce the sanctions, leading to a 'cascade' of obligations. The obligation to pay a fine in (4) is an example of that. In other cases, like in example (5), the sanction is applied automatically. The wedding cake example (6) is a case of mutual obligations. A late delivery is a breach of contract. So the sanction is typically that the customer does not have to pay for the cake. An obligation to pay for the cake may be derived from another rule r or, alternatively, it is given as a fact. Clearly, in the former case we require that $\text{cake}_{sanc} \succ r$. Continuing the previous representations of example (1), (2) and (3), we add the following rules to account for the sanctions in (4), (5) and (6).

inv_{sanc} viol(inv):$t \Rightarrow$ OBLpay_fine:$[t, max]$
pos_{sanc} viol(pos):$t \Rightarrow$ account_blocked:$[t, max]$
wed_{sanc} viol(wed):t, OBLpay:$[t, max] \Rightarrow \neg$OBLpay:$[t+1, max]$

In example (4), suppose that the set of facts is {get_invoice:0}. We have $+\partial$OBLpay:31. The deadline is not respected, and so we derive $+\partial$OBLpay_fine:$[31, max]$. Regarding example (5), suppose that the facts are {open_account:0, $\neg$positive:$[20, 30]$}. We derive $+\partial$OBLpositive:$[0, 30]$. The balance rule is violated, so $+\partial$account_blocked:$[20, max]$. In example (6), suppose we have {order:0, OBLpay:$[0, max]$, wedding:12}. Because the termination rule does not apply, we can derive $+\partial$OBLcake:12. By wed_{viol} we derive $+\partial$viol(wed):12, and therefore $+\partial\neg$OBLpay:$[13, max]$.

Because obligational sanctions are most common, rule inv_{sanc} is generalised into a generic sanction rule, where σ is strongly disliked by the agent being sanctioned.

r_{sanc} viol(r):$t \Rightarrow$ OBLσ:$[t, max]$ (sanction)

[1] Only r_{init} has changed. Alternatively, we could leave r_{init} unchanged, and alter r_{term}.

Notice that sanctions are not always needed. In example (7) we see a non-persistent deadline, with no explicit sanction, apart from the impossibility to achieve one's goals.

(7) Dinner guests must arrive before dinner.

Obligation or Permission? Deadlines are not only connected with obligations. In example (8), there is a permission – or entitlement – which expires at the deadline.

(8) Within 30 days after delivery, customers may return a product to the seller.

Using the principle that an entitlement for one, is an obligation for the other to respect that entitlement, we could also say that after 30 days the seller is no longer obliged to take the product back. Permissions are similar to maintenance obligations, and can be modelled as follows:

ret_{init} delivery:$t \Rightarrow$ PERMreturn_product:$[t, t+30]$

The right to return a product is an example of an achievement permission. Example (9), roughly the opposite of example (2), illustrates a maintenance permission: the right to remain in a certain state for a certain period.

(9) Customers may have a negative balance, for at most 30 consecutive days.

Here we assume to fix the time when the account gets a negative balance.

bal_{init} positive:$t_1 - 1, \neg$positive:$t_1 \Rightarrow$ PERM$\neg$positive:$[t_1, t_1+30]$
bal_{term} $\neg$positive:$[t_1, t_1+30]$, PERM$\neg$positive:$[t_1, t_1+30] \Rightarrow$
$\neg$PERM$\neg$positive:$[t_1+31, max]$
bal_{viol} $\neg$PERM$\neg$positive:$t_2, \neg$positive:$t_2 \Rightarrow$ viol(bal):t_2

Although one may observe that (positive) achievement permissions generally do not have violations or explicit sanctions, violations or sanctions can in fact be connected to maintenance permissions. After the permission ends, usually an original obligation will be restored. Remember that $\neg\text{PERM}\neg\varphi \equiv \text{OBL}\varphi$. So, depending on wether you want to model this as strong or weak permission [9], a negative balance after the permission has run out, does constitute a violation. In some cases, this may also lead to an explicit sanction, as illustrated by (10).

(10) Customers may have a negative balance, for at most 30 consecutive days. After that period, an interest rate of 10% must be paid.

bal_{sanc} viol(bal):$t \Rightarrow$ OBLpay_interest:t

This discussion leads to the following general template rules, for achievement and maintenance permission, respectively. The sanction rule is again r_{sanc}.

r_{init_perm}	$\chi{:}t, \delta{:}t_\delta \Rightarrow \text{PERM}\varphi{:}[t, t_\delta]$	(init. achiev. perm.)
$r_{init\_perm\_main}$	$\chi{:}t, \delta{:}t_\delta \Rightarrow \text{PERM}\varphi{:}[t, t_\delta]$	(init. maint. perm.)
$r_{term\_perm\_main}$	$\chi{:}t, \delta{:}t_\delta, \text{PERM}\varphi{:}[t, t_\delta] \Rightarrow \neg\text{PERM}\varphi{:}[t_\delta+1, max]$	(term. maint. perm.)
$r_{viol\_perm\_main}$	$\neg\text{PERM}\varphi{:}t', \varphi{:}t' \Rightarrow \text{viol}(r){:}t'$	(viol. maint. perm.)

For achievement permissions or entitlements it generally does not make sense to have an explicit sanction. There is only the implicit sanction, namely that the agent is not able to achieve the underlying goals, made possible by the permission. This is a very common situation for permissions, but also for opportunities. In example (11), there is no deontic operator at all. Here, it is the practical opportunity to travel, which expires.

(11) A traveller wants to travel to Rome. The last train to Rome departs at 22:05.

So whence the popular idea that you *must* return the product within 30 days, or that you *must* catch the train? Well, given the fact that you are not satisfied with the product and that you intend to return the product, or given the fact that you intend to travel, the only way to achieve these intentions is before the deadline. The necessity is not deontic, but is based on practical reasoning. We observe that many deadlines have this hybrid nature of deontic and practical reasoning.

This completes our list of parameters. We can now construct different types of deadlines by varying the parameters discussed above. The result is shown in Table 1. For each of the types, we have listed the corresponding examples.

From the table it is clear that some combinations of parameters do not make sense. First, (positive) achievement permissions do not have sanctions, except for the implicit sanction of not being able to achieve one's goal. That excludes type 5, 6 from the table. Moreover, the deadline of a permission or a maintenance obligation, is the moment from which the permission or obligation no longer persist. So by definition, type 5 (again), 7, 9, 11, 13 and 15 are ruled out as well. The remaining 9 deadline types have been

Table 1. Deadline typology

	achieve	operator	sanction	persist	examples	rule templates
1.	Ach.	OBL	Y	Y	payment, fine (4)	$r_{init}, r_{term}, r_{viol}, r_{sanc}$
2.	Ach.	OBL	Y	N	wedding cake (3), (6)	$r_{init\_non\_pers}, r_{term}, r_{viol}, r_{sanc}$
3.	Ach.	OBL	N	Y	payment, no fine (1)	$r_{init}, r_{term}, r_{viol}$
4.	Ach.	OBL	N	N	dinner guests (7)	$r_{init\_non\_pers}, r_{term}, r_{viol}$
5.	Ach.	PERM	Y	Y	no sense	
6.	Ach.	PERM	Y	N	no sense	
7.	Ach.	PERM	N	Y	no sense	
8.	Ach.	PERM	N	N	return product (8)	r_{init_perm}
9.	Maint.	OBL	Y	Y	no sense	
10.	Maint.	OBL	Y	N	account, blocked (5)	$r_{init\_main}, r_{viol\_main}, r_{sanc}$
11.	Maint.	OBL	N	Y	no sense	
12.	Maint.	OBL	N	N	account (2)	$r_{init\_main}, r_{viol\_main}$
13.	Maint.	PERM	Y	Y	no sense	
14.	Maint.	PERM	Y	N	negative, interest (10)	$r_{init\_main\_perm}$, $r_{term\_main\_perm}$, $r_{viol\_main\_perm}$, r_{sanc}
15.	Maint.	PERM	N	Y	no sense	
16.	Maint.	PERM	N	N	negative (9)	$r_{init\_main\_perm}$, $r_{term\_main\_perm}$, $r_{viol\_main\_perm}$

illustrated by examples. By a combination of the template rules for each of these types, we can therefore represent each possible deadline type in a uniform way.

4 Conclusions

In this paper we have given an analysis of deadlines, expressed in Temporal Modal Defeasible Logic. We have chosen a defeasible logic, because it can distinguish exceptions from violations, and because it allows us to have rules override the effect of other rules. This makes it relatively straightforward to model the initiation, termination and violation rules that generally specify a deadline. We have chosen to use intervals, because they provide a natural representation of deadlines, in particular of the distinction between achievement and maintenance obligations.

Although defeasible logic has been already extended to represent contracts [6], this is the first attempt to integrate time. In artificial intelligence, Allen's [1] interval algebra is widely applied. Allen provides a model of the various ways in which temporal intervals may interact and overlap. Note, however, that the satisfiability problem in Allen's algebra is NP-complete [11], which motivated the study of complexity in subalgebras. Although our logic is less expressive and does not cover Allen's algebra operations, its computational complexity is a matter of future research.

Previous work on deadlines typically uses a combination of deontic logic and branching time temporal logic [5,4,3]. However, existing work has typically not addressed what happens *after* the deadline. There are many kinds of deadlines, with different functions. In this paper we have presented a typology of deadlines, using the following parameters: distinction between achievement and maintenance obligations, persistence of the obligation after the deadline, presence of an explicit sanction and the type of operator, obligation or permission. By combining template rules for each of these parameters, we can give a formal characterisation of each of these deadline types. Currently, the typology is validated against a case study of a rental agreement.

References

1. Allen, J.F.: Towards a general theory of action and time. Art. Intelligence 23, 123–154 (1984)
2. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation results for defeasible logic. ACM Transactions on Computational Logic 2(2), 255–287 (2001)
3. Broersen, J., Dignum, F., Dignum, V., Meyer, J.-J.: Designing a deontic logic of deadlines. In: Lomuscio, A.R., Nute, D. (eds.) DEON 2004. LNCS (LNAI), vol. 3065, pp. 43–56. Springer, Heidelberg (2004)
4. Dignum, F., Kuiper, R.: Specifying deadlines with obligations and dense time. International Journal of Electronic Commerce 3(2), 67–85 (1998)
5. Dignum, F., Weigand, H., Verharen, E.: Meeting the deadline: on the formal specification of temporal deontic constraints. In: Michalewicz, M., Raś, Z.W. (eds.) ISMIS 1996. LNCS, vol. 1079, pp. 243–252. Springer, Heidelberg (1996)
6. Governatori, G.: Representing business contracts in RuleML. International Journal of Cooperative Information Systems 14(2-3), 181–216 (2005)
7. Governatori, G., Padmanabhan, V., Rotolo, A.: Rule-based agents in temporalised defeasible logic. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 31–40. Springer, Heidelberg (2006)

8. Governatori, G., Palmirani, M., Riveret, R., Rotolo, A., Sartor, G.: Norm modifications in defeasible logic. In: Proceedings of JURIX 2005, pp. 13–22 (2005)
9. Governatori, G., Rotolo, A., Sartor, G.: Temporalised normative positions in defeasible logic. In: Proceedings of ICAIL 2005, pp. 25–34 (2005)
10. Muller, N.J.: Managing service level agreements. International Journal of Network Management 9(3), 155–166 (1999)
11. Vilain, M., Kautz, H., van Beek, P.: Constraint propagation algorithms for temporal reasoning. In: Readings in qualitative reasoning about physical systems, Morgan Kaufmann, San Francisco (1989)