Timed Dyadic Deontic Logic

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Timed Dyadic Deontic Logic

Karam Younes KHARRAZ [a,1], Martin LEUCKER [a] and Gerardo SCHNEIDER [b]

[a] *ISP, University of Lübeck, Germany*
[b] *University of Gothenburg, Sweden*

Abstract. In this paper, we introduce TDDL, a timed dyadic deontic logic. Our starting point is a version of a dyadic deontic logic with conditional obligations, permissions, and obligations, and with a "reparation" operator for representing contrary-to-duties and contrary-to-prohibitions. We also consider a sequence operator allowing us to define norms as sequences of individual norms and most importantly with timed intervals, allowing us to express deadlines of norms. We provide a trace semantics capturing both satisfaction and violation of norms and discuss fulfillment of TDDL specifications.

Keywords. Normative specification, timed deontic logic

1. Introduction

While the formalization of untimed normative concepts is a quite well-studied topic (though by no means an exhausted nor completely solved issue), less attention has been paid to their combination with (real) time. One of the reasons is that incorporating time to a logic containing modalities for obligations, permissions, and prohibitions—with explicit operators for handling violations—is challenging [1].

That said, the idea of equipping norms with time is not new; see for instance the work by Governatori et al. [6,5,4]. These works, on discrete linear time defeasible deontic logic, prospected important aspects such as the classification of obligations following the timing of their enforcement and the resulting violation rules in a timed setting. In [1], Azzopardi et al. discuss different issues and design choices that need to be considered when adding time to the formalization of normative systems. Their paper is on the challenges regarding expressiveness and computational aspects for both specification and monitoring of timed deontic logics, though concrete solutions are missing.

There is a myriad of deontic logics, and in this paper, we consider *Dyadic Deontic Logic* (or DDL, for short) as a starting point. DDL is a variant of the standard or monadic deontic logic, tackling conditional norms without using material implication. The language of the logic is built on top of atomic propositions alone or put inside of deontic operators. Multiple solutions to arising paradoxes in such logics were proposed using non-monotonic logics or by changing the deontic detachment rule [7,8]. The resulting frameworks come with other limitations as discussed in [9]. We believe that adding time constraints solves certain of those problems of DDL.

---

[1] E-mail: Kharraz@isp.uni-luebeck.de

In this paper, we introduce TDDL, Timed Dyadic Deontic logic, an extension with time of the dyadic deontic logic. Besides the standard operators for (conditional) obligations, permissions, and prohibitions, the underlying logic also has operators for disjunction, sequences, and reparations (to specify penalties in case of violations). Time is explicitly added as intervals to the modalities. We provide trace semantics suitable for conflict (and contradiction) detection as well as for monitoring. More precisely, as models, we consider timed words, i.e., words composed of discrete actions and their timestamps. We provide two semantics relations, one concentrating on duties and prohibitions and the second concentrating on permissions.

The paper is organized as follows. Section 2 introduces TDDL. In Section 3 we briefly discuss the issue of detecting conflicts and contradictions. We discuss related work in Section 4 and we conclude in the last section.

2. TDDL**: Timed dyadic deontic logic**

In this section, we present the logic TDDL. It is based on DDL but restricted to avoid some well-known problems, e.g. our logic does not support *negation* and *conjunction*. At the same time, it is extended to support the notion of discrete-time. Moreover, our logic extends DDL with *preference* and *sequence* operators.

Before presenting the syntax and semantics of our logic, we present norms (clauses) prescribing an online delivery service, which will be used as a motivating example throughout the paper to illustrate the features we want to capture in our logic.

Example 1. Let us consider an online delivery system with the following specification:

- The user is supposed to *collect* the goods when the *home delivery* shows up. The date of the home delivery is fixed between three and five days after the order has been issued. If the user does not collect the goods on the day of the home delivery, the post agent deposit leaves a missed delivery notice and the user is supposed to *collect* the goods from the closest post station within 7 days after receiving the notice.
- The user may *return* the collected goods within 30 days of the delivery.

Let us look at this example more closely: We have individual *agents* mentioned in the clauses. However, norms concern mostly one of them, the service user. The other agent, the post agent is barely mentioned. In general, the agents perform some *actions* such as delivering goods, returning goods, etc. The only active action from the post agent is to leave a missed delivery notice, which may be seen as a condition to enforce the norm for picking the goods from the post. Note also that there is a notion of *preference*: picking up the goods on the day of the delivery is the normal *desired* behavior from the agent while picking up the goods from the post is considered as *reparation*. There is also a temporal order between the reparations and the desired behavior. Most of these actions underly temporal constraints, meaning they should occur within a certain time interval, like collecting goods within 7 days after receiving a notice or returning goods within 30 days of delivery. Notice that the corresponding intervals are often relative to other actions, like 30 days after delivery. In the next subsection, we use these observations to distill our logic.

$$\mathtt{N} := Dop^I(a) \mid Dop^{I_2}(a \mid^{I_1} b) \text{ with } Dop \in \{O \mid P \mid F\}$$

$$\mathtt{NC} := \mathtt{N} \mid \mathtt{N} \ op \ \mathtt{N} \ \text{ with } op \in \{\gg, ;, \vee\}$$

$$\mathtt{NS} := \{\mathtt{NC}_1, \mathtt{NC}_2 \dots, \mathtt{NC}_n\}$$

Figure 1. Syntax of TDDL

2.1. TDDL *Syntax*

The syntax of TDDL is shown in Figure 1, norms are formed using deontic operators, actions and time intervals. The deontic operators are: Obliged *O*, Forbidden *F*, and Permitted *P*. Since we do not have negation we need the three modalities. Actions are from a set Σ that consist of all possible discrete actions (from the agent and the "environment").[2]

We assume actions are atomic, meaning their duration is a one-time step and that two actions cannot happen at the same time step. Time intervals are defined from the domain $\mathbb{I}^+ = [0, +\infty[$. An interval is formed by a pair, or union of pairs, $[i, s]$, with $i, s \in \mathbb{N}$ and $0 \leq i \leq s$. As in DDL, norms comes in two flavors: *monadic* or *dyadic*. A monadic norm is formed with one action and one interval. For instance, $O^{[0,4]}(coll)$ means that the agent has to *achieve* to collect the goods within 4 days. $F(open\_door) \equiv F^{[0,+\infty[}(open\_door)$ means that the agent is *always* forbidden to open the door; $P^{[0,30]}(ret)$ means that the agent has the right to return goods between 0 and 30 time steps.[3] Dyadic operators take two actions and two intervals. The left action is the action concerning the agent whilst the right one is the triggering action coming possibly from the environment. The two intervals are respectively the *norm validity interval* and the *reactivity interval*. For instance, $O^I(a \mid^R b)$ means that the agent is obliged to react performing *a* within the reactivity interval *R after* the environment had done action *b* within interval *I*.

Norms may be composed using the operators of preference "$\gg$" and sequence ";".

Sequence We use this operator to specify a linear order between the fulfillment of norms. For example, $\mathtt{NC}_3 := O^{[3,5]}(coll); P^{[0,30]}(ret)$ specifies that to fulfill $\mathtt{NC}_3$, the agent has to fulfill the collection before fulfilling the permission to return the goods. The interval of the right norm is relative to the left norm. Another sequence operator could be specified for cases requiring having absolute intervals in the second norm, but is not considered here due to space limitations.

Preference Unlike the Kripke semantics of DDL, we do not encode the preference relation in the model of the logic. For example, $\mathtt{NC}_4 := O^{[3,5]}(h\_coll) \gg O(coll \mid^{[0,7]} p\_del)$, prescribes that collecting the goods at home is *more preferable* than collecting it from the post after receiving the failed home collection notice from the post agent. Note that this operator is not symmetric unlike the logical or. Like the sequence, the preference operator could have a variant where the second interval is interpreted as absolute, but again this is left for a full version of the paper.

---

[2]W.l.o.g. and for simplicity, we assume that the contract concerns only one agent "against" an environment. This is a question more of terminology for presenting our ideas: we can also consider, as usually done in the literature, that actions encode the active agent/user performing the action (or to which the norm applies to).

[3]We talk in general about "time steps" with the understanding that it might mean different time units depending on the context (e.g., here it might mean "days".)

Normative system   A normative system is a set of composite or simple norms.

Example 2. The norms of example 1 are specified in TDDL as a composed norm:

$$\texttt{NC}_{\texttt{Delivery}} := (O^{[3,5]}(h\_coll) \gg O(coll \,|^{[0,7]} \, p\_del)); P^{[0,30]}(ret).$$

2.2. Duty and Right trace semantics

One special feature compared to DDL is that we define two different satisfaction relations in TDDL, $\vDash_D$ (*duty*) and $\vDash_R$ (*right*). The duty relation means "did the agent fulfill her duties". A duty implies performing an action in the case of an obligation or avoiding it in the case of a prohibition. The right relation gives the answer to the question: "did the agent used her right?". For a normative system, our model is a word $w = (p_1, \tau_1) \dots (p_n, \tau_n)$ where the actions are from the agent and the system (environment): $p_i \in \Sigma$ and $\tau_i \in \mathbb{N}$ are timestamps. Let us first concentrate on the duty semantics.

Obligations and prohibitions   To fulfill an obligation, it is enough to have one occurrence of the specified action. For prohibitions, one occurrence inside the scope of the prohibition violates the duty semantics. Thus, we define the satisfaction relations for the monadic operators as

$$w \vDash_D \; O^I(a) \text{ iff } \exists t \in I.a = w(t)$$

$$w \vDash_D \; F^I(a) \text{ iff } \forall t \in I.a \neq w(t)$$

For the dyadic operators, the semantics is given as:

$$w \vDash_D \; O^I(a|^R b) \text{ iff } \forall t \in I.b \notin w(t) \text{ or } (\exists min(t) \in I.w(t) = b \text{ and } w \vDash_D O^{R+t}(a))$$

$$w \vDash_D \; F^I(a|^R b) \text{ iff } \forall t \in I.b \notin w(t) \text{ or } (\forall t \in I.w(t) = b \rightarrow w \vDash_D F^{R+t}(a))$$

For example, we have that $(coll, 3) \vDash_D \; O^{[3,5]}()$, $(h\_coll, 3) \nvDash_D \; F^{[3,5]}(coll)$ and $(p\_del, 4)(coll, 6) \vDash_D \; O(coll|^{[0,7]} p\_del)$.

Permissions in the duty semantics   We define permissions in the most simple way, where we say that a permission is not concerned by the duty semantics. Hence, we define:

$$w \vDash_D \; P^I(a) \text{ always}$$

$$w \vDash_D \; P^I(a|^R b) \text{ always}$$

Another possible kind of operator is the *strict permission*, where using a right out of the context when the conditions are satisfied could be interpreted as a violation of the duty semantics.[4]

Composed norms   For the sequence and the preference operators, we have to know when exactly the first, left norm has been fulfilled or violated because the fulfillment or reparation by the right norm is *relative* to the one in the left. This information is provided by the two functions *violation prefix* VP and *fulfillment prefix* FP which provide for a given norm and trace the prefix that violates or fulfills the norm, respectively, and is defined further below. We now define the semantics for the composed operators as:

---

[4]A *strict permission*, SP, could be defined as $SP^I(a) \equiv \{P^I(a), F^{[0,+\infty]-I}(a)\}$.

$$w \vDash_D \text{NC}_1^{I_1} \gg \text{NC}_2^{I_2} \quad \text{iff} \quad w \vDash_D \text{NC}_1^{I_1} \ \ or \ \ (w \nvDash_D \text{NC}_1^{I_1} \ and \ w \vDash_D \text{NC}_2^{I_2 + VP(w, NC_1)})$$

$$w \vDash_D \text{NC}_1^{I_1} \ ; \ \text{NC}_2^{I_2} \quad \text{iff} \quad w \vDash_D \text{NC}_1^{I_1} \ \ and \ \ w \vDash_D \text{NC}^{I_2 + \text{FP}(w, NC_1)}$$

$$w \vDash_D \text{NC}_1^{I_1} \vee \text{NC}_2^{I_2} \quad \text{iff} \quad w \vDash_D \text{NC}_1^{I_1} \ \ or \ \ w \vDash_D \text{NC}^{I_2}$$

Remark 1. To be able to capture the nuance of fulfilling a composed norm with preference, one can add an index to the duty satisfaction relation i.e $\vDash_{D,1}$ to express the fact that the norm was fulfilled in the "best settings" and $\vDash_{D,2}$ for the second best setting.

Right semantics   Obligations and permissions are not concerned with the right satisfaction relation. For permissions, one or more occurrences of the concerned action indicate that the right has been used. So, we have for instance that $(ret, 3) \vDash_R P^{[0,30]}(ret)$ but $(ret, 3) \nvDash_R O^{[0,30]}(ret)$. Note that the semantics of composed norms with the preference operator is not defined. The right semantics is defined as follows:

$$w \nvDash_R DOP^I(a) \text{ iff } DOP \in \{O, F\}$$

$$w \nvDash_R DOP^I(a|^R b) \text{ iff } DOP \in \{O, F\}$$

$$w \vDash_R P^I(a) \text{ iff } \exists t \in I . w(t) = a$$

$$w \vDash_R P^I(a|^R b) \text{ iff } \exists t' \in I . w(t) = b \text{ and } \exists w \vDash_R P^{R+t}(a)$$

$$w \vDash_R \text{NC}_1^{I_1} \ ; \ \text{NC}_2^{I_2} \text{ iff } w \vDash_R \text{NC}_1^{I_1} \text{ or } w \vDash_R \text{NC}^{I_2 + \text{FP}(w, NC_1)}$$

$$w \vDash_R \text{NC}_1^{I_1} \ \vee \ \text{NC}_2^{I_2} \text{ iff } w \vDash_R \text{NC}_1^{I_1} \text{ or } w \vDash_R \text{NC}^{I_2}$$

Normative systems   A normative system is satisfied according to the duty semantics if all norms are satisfied within this relation. On the other hand, we only require that at least one norm in the system satisfies the right semantics for the whole normative system to satisfy it:

It remains to define the two functions VP and FP that identify the earliest timestamps (shortest prefix) for which a norm is violated or fulfilled.

Monadic operators   The violation prefix of monadic obligations is the maximum element of the interval, while that of a prohibition is the first occurrence of the forbidden action. There is no violation prefix for permissions. We thus have:

$$\text{VP}(w, O^I(a)) := max(I) \qquad \text{iff } w \nvDash_D O^I(a)$$

$$\text{VP}(w, F^I(a)) := t \in I \qquad \text{iff } w_{0,t} \nvDash_D F^I(a) \text{ and } \nexists t' < t . w_{0, t'} \nvDash_D F^I(a)$$

So, $\text{VP}((h\_coll, 7), O^{[3,5]}(h\_coll)) = 5$ and $\text{VP}((h\_coll, 2), F^{[3,5]}(h\_coll)) = 2$.

Dyadic operators   The violation prefixes for the dyadic operators are more complex. In the case of obligations, it is set to the maximum element of the reactive interval updated with the timestamp of the first occurrence of the triggering action. For a prohibition, this prefix is the first occurrence of the forbidden action:

$$\text{VP}(w, O^I(a|^R b)) := t + max(R) \quad \text{iff } t = first_{occ}(w, b) \text{ and } w \nvDash_D O^I(a|^R b)$$

$$\text{VP}(w, F^I(a|^R b)) := t \qquad \text{iff } w(t) = a \text{ and } w_{0, (t-1)} \nvDash_D F^I(a|^R b) \text{ and } w \nvDash_D F^I(a|^R b)$$

For instance, we have $\text{VP}(\{(p\_del, 7), (ret, 15)\}, O(p\_del|^{[0,7]} coll)) = 14$.

Composed norms   The violation prefix for the preference operator is the violation of the right norm updated with the violation prefix of the left norm. For sequences, the violation prefix could have two forms: the violation prefix of the left norm or the violation prefix of the right norm updated with the fulfillment prefix of the first norm. This gives rise to:

$$\text{VP}(w, \text{NC}_1^{I_1} \gg \text{NC}_2^{I_2}) := \text{VP}(w, \text{NC}_2^{I_{\text{VP1}}})$$

$$\text{iff } I_{\text{VP1}} = I_2 + \text{VP}(w, \text{NC}_1^{I_1}) \text{ and } w \nvDash_D \text{NC}_1^{I_1} \gg \text{NC}_2^{I_2}$$

$$\text{VP}(w, \text{NC}_1^{I_1} \text{ ; } \text{NC}_2^{I_2}) := \begin{cases} \text{VP}(w, \text{NC}_1^{I_1}) \text{ iff } w \nvDash_D \text{NC}_1^{I_1} \\ \text{VP}(w, \text{NC}_2^{I_2 + \text{FP}w, \text{NC}_1^{I_1}}) \text{ iff } w \nvDash_D \text{NC}_1^{I_1} \text{ ; } \text{NC}_2^{I_2} \end{cases}$$

$$\text{VP}(w, \text{NC}_1^{I_1} \vee \text{NC}_2^{I_2}) := max((\text{VP}(w, \text{NC}_1^{I_1}), \text{VP}(w, \text{NC}_2^{I_2}))).$$

For example: $\text{VP}((p\_del, 7)(ret, 15), (O^{[3,5]}(h\_coll) \gg O(coll|^{[0,7]}p\_del))) = 14$, and $\text{VP}((p\_delevery, 7)(ret, 15), (O(p\_del|^{[0,7]}coll); P^{[0,30]}(ret)) = 14$.

Fulfillment prefixes   An obligation is satisfied by the first occurrence of the corresponding action (respecting the time interval). For prohibitions, the fulfillment prefix is the limit of the interval. Fulfillment in our paper refers to the possibility of exercising rights and achieving duties:

$$\text{FP}(w, O^I(a)) := t \in I \qquad \text{iff } w_{0,t} \vDash_D O^I(a) \text{ and } (\forall t' < t. w_{0,t'} \nvDash_D O^I(a))$$

$$\text{FP}(w, F^I(a)) := max(I) \qquad \text{iff } w \nvDash_D F^I(a)$$

$$\text{FP}(w, O^I(a|^R b)) := \begin{cases} max(I) \text{ iff } \nexists t' \in I. w(t') = b \\ t \text{ iff } t' = first_{oc}(w, b) \text{ and } t = first_{oc}(w^{t'}, a) \end{cases}$$

$$\text{FP}(w, F^I(a|^R b)) := \begin{cases} max(I) \text{ iff } \nexists t' \in I. w(t') = b \\ t \text{ iff } t' = last_{oc}(w_{0,max(I)}, b) \text{ and } t = t' + R \end{cases}$$

For permissions, the prefix depends on whether the right has been used or not. If the right was not used then the fulfillment prefix is set up to be the maximum element of the interval. In our example, $\text{FP}(\{(coll, 3)\}, O^{[3,5]}(coll)) = 3$ and $\text{FP}(\{(coll, 1), (coll, 7)\}, F^{[3,5]}(coll)) = 5$. Formally,

$$\text{FP}(w, P^I(a)) := \begin{cases} t \text{ iff } t = first_{oc}(w_I, a) \\ max(I) \text{ iff } w \nvDash_R P^I(a) \end{cases}$$

$$\text{FP}(w, P^I(a|^R b)) := \begin{cases} max(I) \text{ iff } w \nvDash_r P^I(a|^R b) \\ t \text{ iff } t = first_{oc}(w_{I+R}, a) \text{ and } \\ \qquad \exists t' in [t - max(R), t - min(R)]. w(t') = b \end{cases}$$

So, $\text{FP}(\{(ret, 3)\}, P^{[0,30]}(ret)) = 3$ and $\text{FP}(\{(coll, 32)\}, P^{[0,30]}(ret)) = 30$.

The fulfillment prefix for sequences is the fulfillment prefix of the right norm updated with the fulfillment prefix of the left norm. On the other hand, for composing norms with preference operators, we consider two cases, the second case is relative to the violation prefix of the left norm. The formalization is as follows:

$$\text{FP}(w, \text{NC}_1^{I_1} \text{ ; } \text{NC}_2^{I_2}) := \text{FP}(w, \text{NC}_2^{I_{FP}}) \quad \text{iff } I_{FP} = I_2 + \text{FP}(w, \text{NC}_1^{I_1}) \text{ and } w \vDash_D \text{NC}_1^{I_1} \text{ ; } \text{NC}_2^{I_2}$$

$$\text{FP}(w, \text{NC}_1^{I_1} \vee \text{NC}_2^{I_2}) := x \qquad\qquad \text{iff } x = min((\text{FP}(w, \text{NC}_1^{I_1}), \text{FP}(w, \text{NC}_2^{I_2}))$$

$$\text{FP}(w, \text{NC}_1^{I_1} \gg \text{NC}_2^{I_2}) := \begin{cases} \text{FP}(w, \text{NC}_1^{I_1}) \text{ iff } w \vDash_D \text{NC}_1^{I_1} \\ \text{FP}(w, \text{NC}_2^{I_{FP1}}) \text{ iff } I_{FP1} = I_2 + \text{VP}(w, \text{NC}_1^{I_1}) \text{ and } w \vDash_D \text{NC}_1^{I_1} \gg \text{NC}_2^{I_2} \end{cases}$$

Example 3. Let us consider the following traces of $\text{NC}_{\text{Delivery}}$:
$w_1 := (h\_coll, 3)(ret, 31); \quad w_2 := (p\_del, 5)(coll, 18); \quad w_3 := (h\_coll, 3)(ret, 37)$ We
can see that: $\text{FP}(w_1, (O^{[3,5]}(h\_coll) \gg O(coll \,|^{[0,7]} p\_del)) = 3$
$\text{VP}(w_2, (O^{[3,5]}(h\_coll) \gg O(coll \,|^{[0,7]} p\_del)) = 12$
$w_1 \vDash_D \text{NC}_{\text{Delivery}} \text{ and } w_1 \vDash_R \text{NC}_{\text{Delivery}} \text{ and } w_2 \nvDash_D \text{NC}_{\text{Delivery}} \text{ and } w_2 \nvDash_R \text{NC}_{\text{Delivery}}$
$w_3 \vDash_D \text{NC}_{\text{Delivery}} \text{ and } w_2 \nvDash_R \text{NC}_{\text{Delivery}}$

3. Fulfillabilty of normative systems in TDDL

Fulfillability is a important aspect of Normative systems. Following the aforementioned
semantics it means that it is possible to satisfy both in the right and duty semantics. Es-
pecially specifying complex norms is a difficult task and automatic sanity checks would
be desirable. We consider two notions important for such a sanity check: contradiction
and conflict.

We say that two norms are *conflicting* if there is situation where it is not possible to
use a right without breaching another norm. We call two norms *contradicting* when there
is no possible trace that can fulfill the duties of all the norms.

For example, $\text{NS}_1 := \{O^{[3,5]}(a), F^{[2,6]}(a)\}$ is contradicting because for each time
stamp where the first norm could be fulfilled (namely 3,4,5), the second would be
breached. For $\text{NS}_2 := \{P^{[3,5]}(a), F^{[4,5]}(a)\}$ there exists timestamps (4,5) yielding to a vi-
olation if the agent uses his right thus leading to a conflict. Not using the permission
would satisfy the norms in the duty semantic but not the right one, so the two norms
are conflicting in the interval $[4,5]$. An algorithm to analyze a normative system and to
detect contradictions and conflicts, signaling the unfulfillabilty of the concerned part of
the original system, will be presented on an extended version of this paper.

4. Related Work

Partial normative specifications with time have been given by Governatori et al., for in-
stance in [6,5,4]. The formalization consists of using *defeasible* and *defeators* rules to
initialize, terminate and define violations of norms. The timed settings are intervals in $\mathbb{N}$.
In those works timed actions are of different kinds: *achievement*, *maintenance* and *punc-
tual*. In our work obligations and permissions are *achievements*, while prohibitions are
of *maintenance* type. We do not support punctual since we do not allow the occurrence
of simultaneous actions.

A "deontic calculus" extended with time intervals has been presented in [2,3]. The
main difference with our work is that we are here considering a logic instead of a calcu-
lus: we have a formal (denotational) semantics defined in terms of a satisfaction relation
over models, while they provide an operation view of the syntax (calculus).

Given the difference in the formalization and the intention (defeasible logic and
calculi), it is difficult to make a concrete comparison with our approach except for the
fact that we provide a deontic logic with explicit time.

5. Conclusion

In this paper, we presented a first suggestion of a timed dyadic deontic logic allowing to reason over prohibitions, obligations and permissions, within certain *timed intervals*. To support the different flavor of obligations and prohibition on one hand and permissions on the other, our logic comes with two different semantics relations, a first concentrating on *duties* while the second indicates which *rights* have been used when looking at action sequences that should adhere to the given norm.

It is very easy to define norms that have inherent conflicts or are even contradicting, for example by specifying overlapping intervals in which certain actions are both forbidden and obligated. We defined these notions formally and indicated that algorithmic support for improving such specifications can be done as future work.

Our work may be enhanced in several directions. The current expressiveness can be extended by further operators and by giving both relative and absolute notions of timed intervals. Also, the analysis of given norms may be enriched by providing checks for conflicts and contradictions. We leave such extensions for a full version of the paper.

References

[1] Azzopardi, S., Pace, G., Schapachnik, F., Schneider, G.: On the specification and monitoring of timed normative systems. In: RV'21. LNCS, Springer (2021)

[2] Cambronero, M.E., Llana, L., Pace, G.J.: A calculus supporting contract reasoning and monitoring. IEEE Access **5**, 6735–6745 (2017)

[3] García, A.A., Cambronero, M., Colombo, C., Llana, L., Pace, G.J.: Runtime verification of contracts with themulus. In: SEFM'20. LNCS, vol. 12310, pp. 231–246. Springer (2020).

[4] Governatori, G., Hulstijn, J., Riveret, R., Rotolo, A.: Characterising deadlines in temporal modal defeasible logic. In: Australasian Joint Conference on Artificial Intelligence. pp. 486–496. Springer (2007)

[5] Governatori, G., Rotolo, A.: Justice delayed is justice denied: Logics for a temporal account of reparations and legal compliance. In: CLIMA'11. pp. 364–382. Springer (2011)

[6] Governatori, G., Rotolo, A., Sartor, G.: Temporalised normative positions in defeasible logic. In: ICAIL'05. pp. 25–34 (2005)

[7] Horty, J.F.: Nonmonotonic foundations for deontic logic. In: Defeasible deontic logic, pp. 17–44. Springer (1997)

[8] Parent, X., van der Torre, L.: Aggregative deontic detachment for normative reasoning. In: KR'14. AAAI Press (2014)

[9] Pigozzi, G., van der Torre, L.: Multiagent deontic logic and its challenges from a normative systems perspective. IfCoLog Journal of Logics and Their Applications pp. 2929–2993 (2017)