On Conflicts and Satisfiability in Metric Timed Normative Logics

Legal Knowledge and Information Systems
J. Savelka et al. (Eds.)

doi:10.3233/FAIA241260

On Conflicts and Satisfiability in Metric Timed Normative Logics

Karam KHARRAZ [a,1], Gerardo SCHNEIDER [b] and Martin LEUCKER [a]
[a] *University of Lübeck, Lübeck, Germany*
[b] *University of Gothenburg, Sweden*

Abstract. In this paper, we study the concept of *conflict* in the setting of timed normative logical specification languages. To this end, we introduce the Flat Monadic Metric Time Normative Logic suitable for specifying the behavior of *basic* timed normative systems using sets of intervals. We provide a characterization of normative conflicts by the satisfiability of the formula and its sub-formulas. Moreover, an SMT-based satisfiability procedure for FMMTNL is provided.

Keywords. Normative systems, Timed logic, SMT solving, Conflict analysis

1. Introduction

Among the properties needed for normative formalisms comes the notion of normative conflict and its detection as introduced in the seminal work by Sartor [1992]. This is required during the design phase of a new normative system, when introducing new rules to a normative system, or when composing multiple normative systems together. In a *metric-timed* setting, norms are defined with specific time constraints like deadlines or clock-based conditions. Conflicts arise at exact moments when a norm both permits and prohibits the same action. In contrast, in *logical time*, conflicts resemble contradictions, making norms impossible to be satisfied entirely. On the other hand, in metric time conflicts arise at specific time points, which do not lead necessarily to contradictions. While there are several metric time normative formalisms (Kharraz et al. [2021]; Governatori et al. [2007]; Hvitved et al. [2012]; Camilleri et al. [2014]; Farmer and Hu [2016]), their main focus is to demonstrate the use of metric time to express additional aspects such as deadlines and reparations. Conflicts may be resolved by the use of defeasibility (as in Governatori and Rotolo [2011]), or by using a revision operator to rewrite the time intervals as in Tamargo et al. [2019]. Other approaches resolve conflicts via automata transformation (e.g., Fenech et al. [2009]). That said, a clear characterization of *timed conflicts* is lacking, as discussed in Azzopardi et al. [2021].

The purpose of this paper is to provide a formal account of normative conflicts and their relation to the satisfiability of logical formulas. To do so, we introduce a simple normative metric timed logic, which we claim can be considered a key sub-fragment of most metric time formalisms, to characterize these concepts. We explore the satisfiability of

[1]Corresponding Author: Kharraz@isp.uni-luebeck.de

our logic, showing it to be NP-complete and offer a satisfiability procedure by translating the problem into a corresponding SMT problem. We define different types of normative conflicts and show their connection to the satisfiability of a formula in our logic.

2. Preliminaries

We consider $\mathbb{N}$ to represent all positive integers including 0, which will be the base for the (discrete) notion of time. An interval I is a subset of natural numbers $\mathbb{N}$ formed by a *min* (minimum) from $\mathbb{N}$ and a *max* (maximum) from $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$, a valid interval $I = [t_{min}, t_{max}]$ is defined with $t_{max} \geq t_{min}$ to refer to a subset of elements from $\mathbb{N}$ less or equal than t_{max} and greater or equal to t_{min}. For $\mathbb{N}^\infty$ the order relation has two rules: $(rule1)$ $t_{min} <_\infty t_{max}$ if $t_{min} < t_{max}$ and $t_{min}, t_{max} \in \mathbb{N}$; $(rule2)$ $t_{min} <_\infty \infty$ if $t_{min} \in \mathbb{N}$. The domain of intervals $\mathbb{I}$ is defined as: $\mathbb{I} := \{[t_{min}, t_{max}] \mid t_{min} \in \mathbb{N} \wedge t_{max} \in \mathbb{N}^\infty\}$.

An *interval set*, written $\mathfrak{I}$ is a finite set of intervals from $2^{\mathbb{I}}$ such that its elements are ordered according to precedes relation ($\prec$) from the algebra in Allen [1983]. The relation is defined as: $[a,b] \prec [c,d]$ iff $b < c$. We have $(I_i \in \mathfrak{I}$ and $I_j \in \mathfrak{I}) \implies (I_j \prec I_i$ or $I_i \prec I_j)$.

An interval I is a *sub-element* of $\mathfrak{I}$ (written $I \underline{\in} \mathfrak{I}$) if it is included in one of the elements of the set of intervals. we write $I \underline{\in} \mathfrak{I}$ iff $\exists I' \in \mathfrak{I} : I \subseteq I'$. We abuse the notation for time points $t \in \mathbb{N}$ and write $t \underline{\in} \mathfrak{I}$ iff $[t,t] \underline{\in} \mathfrak{I}$. We say $\mathfrak{I}$ is *included* in $\mathfrak{I}'$ and write $\mathfrak{I} \subset \mathfrak{I}'$ iff $\forall I \in \mathfrak{I}\ \exists I' \in \mathfrak{I}' : I \subset I'$.

Metric time models. Models in this work are timed traces, they represent the behavior of an agent/system. We assume that an agent/system can only perform one action per time point from $\mathbb{N}$. A *finite timed trace* τ is defined as a finite sequence of timed events, $\tau \in (\Sigma \times \mathbb{N})^*$, where the time stamps of the events are ordered and relative to the same *global clock*: $\tau := \langle (a_1, t_1), (a_2, t_2), \ldots, (a_n, t_n) \rangle$. A *timed event* is a tuple from $\Sigma \times \mathbb{N}$, *e.g.*, $(open\_door, 15)$ means that the action *open_door* happened at time-point 15. *Actions* are defined over a finite alphabet of actions $\Sigma = \{a_1, \ldots, a_n\}$. In our settings, actions are *atomic*, meaning their duration is punctual from $\mathbb{N}$. At a time stamp t, at most one action can be performed by an agent. We refer to the occurrence of an action a at a time point $t \in \mathbb{N}$ by a tuple (a,t). The partial function $\rho : (\Sigma \times \mathbb{N})^* \times \mathbb{N} \rightharpoonup \Sigma$ returns the action in the input trace τ at time t, if and only if it exists.

3. The Flat Monadic Metric Time Normative Logic (FMMTNL)

The *flat monadic metric time normative logic*, written FMMTNL, is called *flat* because it has no temporal operators; and *monadic* because there is no operator to specify conditions. An obligation is written O and prohibition is written F. Unlike other metric time formalisms, we use sets of intervals to specify norms to increase the conciseness of formulas. The logic has *the mininal operators* such as conjunction $\sqcap$, disjunction $\sqcup$, and negation $\neg$.

Definition 1 Syntax of FMMTNL. *The syntax of FMMTNL is defined recursively as:*

$$\begin{array}{lcl} \phi & ::= & O^{\mathfrak{I}}(a) \mid F^{\mathfrak{I}}(a) \mid \phi \sqcup \phi \mid \phi \sqcap \phi \mid \neg\phi \\ \mathfrak{I} & ::= & \{I, \ldots, I\} \\ I & ::= & [t_{min}, t_{max}] \end{array}$$

The trace semantics in this paper will be referred to as the *duty semantics*: The satisfaction relation is defined as $\vDash$: $(\Sigma \times \mathbb{N})^* \times \mathsf{FMMTNL}$ capturing the fact that a timed finite trace τ representing an agent's behavior satisfies *the duties* specified by a formula ϕ representing a normative system. The duty semantics defines how to satisfy obligations that are required to occur at least once within the set of intervals, commonly known as *achievement obligations* in Dignum and Kuiper [1998]. Prohibition forbids any occurrence of an action within the specified set of intervals, which could be seen as a *negative achievement* over a set of time points. Conjunction ($\sqcap$) adds more duties and disjunction ($\sqcup$) offers the agent the choice of a duty instead of another. Finally, we specify the negation $\neg$ in terms of the non-satisfaction of the formula.

Definition 2. *The duty semantics is defined recursively on the structure of the formula:*

$$\tau \vDash O^{\mathfrak{I}}(a) \textit{ iff } \exists t \Subset \mathfrak{I} : \rho(\tau,t) = a, \quad \tau \vDash F^{\mathfrak{I}}(a) \textit{ iff } \forall t \Subset \mathfrak{I} : \rho(\tau,t) \neq a,$$
$$\tau \vDash \phi \sqcup \phi' \textit{ iff } \tau \vDash \phi \textit{ or } \tau \vDash \phi', \quad \tau \vDash \phi \sqcap \phi' \textit{ iff } \tau \vDash \phi \textit{ and } \tau \vDash \phi',$$
$$\tau \vDash \neg\phi \textit{ iff } \tau \nvDash \phi.$$

Definition 3 Satisfiability. *A formula ϕ is* satisfiable *or in short* sat *iff there exists a trace satisfying the formula ϕ: ϕ is* sat *iff* $\exists \tau : \tau \vDash \phi$

SMT solving We implemented a sound satisfiability checker using an SMT solver written in Python (we used the Z3 SMT solver library).[2] The implementation is based on the function synthetize below that uses the solver capability of the SMT tool and returns a model τ_{min} when the formula is satisfiable. The SAT problem is decomposed into three formulas: $\mathsf{synthetize}(\phi) := solve\big[\mathsf{CstrC}(\phi, none) \wedge Disjoint(\tau_{min}) \wedge \big(|\tau_{min}| \leq \eta(\phi)\big)\big]$.

Where *Disjoint* encodes the conditions on valid traces as defined in Section 2 where at any time point, only one action could be performed. $Disjoint(\tau_{min}) := \forall\{(act_1,t_1),(act_2,t_2)\} \subseteq \tau_{min} : t_2 \neq t_1$. The constraint $|\tau_{min}| \leq \eta(\phi)$ encodes Lemma 3.1 to soundly bound the size of models to satisfy a formula and get the *minimal* model τ_{min}.

Lemma 3.1. *Let $\eta(\phi)$ be the number of obligations in a formula ϕ. When the formula ϕ is satisfiable there exists a minimal trace w' with at most $\eta(\phi)$ elements that satisfy the formula ϕ, i.e:* $(\exists \tau : \tau \vDash \phi) \implies \big(\exists \tau_{min} : |\tau_{min}| \leq \eta(\phi) \textit{ and } \tau_{min} \vDash \phi\big)$.

The constraint collection function CstrC encodes the duty semantics in the SMT theory of sets over $(\Sigma \times \mathbb{N})$. The function recursively extracts constraints and links them using the logical operators $\wedge$ and $\vee$. CstrC is recursively defined as follows:

$$\mathsf{CstrC}(\phi) := \begin{cases} \exists (a,t_s) \in \tau_{min} : t_s \Subset \mathfrak{I} & \text{if } \phi = O^{\mathfrak{I}}(a), \\ \forall (x,t) \in \tau_{min} : (x = a \Rightarrow t \notin \mathfrak{I}) & \text{if } \phi = F^{\mathfrak{I}}(a), \\ \mathsf{CstrC}(\phi_1) \wedge \mathsf{CstrC}(\phi_2) & \text{if } \phi = \phi_1 \sqcap \phi_2, \\ \mathsf{CstrC}(\phi_1) \vee \mathsf{CstrC}(\phi_2) & \text{if } \phi = \phi_1 \sqcup \phi_2. \end{cases}$$

For obligation $O^{\mathfrak{I}}(a)$, the condition on the minimal trace is to contain an event (a,t_s) where t_s is the solution time point returned. For prohibition $F^{\mathfrak{I}}(a)$, the condition on the

[2] https://github.com/khrrzkrm/FMMTNL_Solver/

minimal trace is that for all events where the timestamp is in the forbidden interval, the action is different from *a*. The constraints are linked using the logical equivalent operators $\wedge, \vee$ for conjunction and disjunction.

Theorem 1. *The satisfiability problem of* FMMTNL *is NP-complete.*

4. Relating normative conflicts to satisfiability

After defining the logic and studying satisfiability, we classify in this section normative conflicts as *ontic* and *deontic*. Compared to Colombo Tosatto et al. [2014], which includes more obligation types (e.g., maintenance, preemptive). On the other hand, our approach is more precise by distinguishing partial and total time conflicts. We also account for action costs in ontic conflicts in addition to interval overlapping. In the following definitions, we assume that ϕ is in Conjunctive Normal Form (CNF) and that the *contains* relation is evaluated at the clause level.

Definition 4 Deontic time conflict (DC)**.** *We define a deontic conflict function written* DC*, returning true when for a formula ϕ, at time point t, an action a is both obliged and forbidden.* $\mathsf{DC} : \mathsf{FMMTNL} \times \mathbb{N} \times \Sigma \to \{\top, \bot\}$ *:*

$$\mathsf{DC}(\phi, t, a) = \begin{cases} \top \text{ if } \phi \text{ contains } F^{\mathfrak{I}_1}(a) \sqcap O^{\mathfrak{I}_2}(a) \text{ and } t \Subset \mathfrak{I}_1 \cap \mathfrak{I}_2, \\ \bot \text{ otherwise .} \end{cases}$$

A formula ϕ *has a deontic timed conflict* iff there exists a time point t that is a deontic timed conflict: ϕ has a deontic conflict iff $\exists(a,t) : \mathsf{DC}(\phi, a, t) = \top$.

Definition 5 Ontic time conflict. *A formula ϕ* has an ontic conflict *iff the formula contains a set of n obligations on different actions on the same set of intervals with m time points where n > m. That is,* ϕ has an ontic conflict *iff ϕ contains $O^{\mathfrak{I}}(a_1) \sqcap O^{\mathfrak{I}}(a_2) \sqcap \cdots \sqcap O^{\mathfrak{I}}(a_n)$ and $|\mathfrak{I}| < n-1$.*

We have the following important results relating conflict and (un)satisfiability.

Lemma 4.1. *In* FMMTNL*, if ϕ is unsatisfiable then ϕ has a deontic or an ontic conflict.*

Lemma 4.2. *In* FMMTNL*, if ϕ has an ontic conflict then ϕ is unsatisfiable.*

Note that a formula with a deontic time conflict is not necessarily unsatisfiable, which leads us to investigate an additional sub-characterization, we define *total deontic timed conflict* when the formula is unsatisfiable and *partial time deontic conflict* which we relate to *unambiguous satisfiability*. This property helps for the refinement of a normative system and thus simplifies the scheduling process for an agent under the normative systems as it narrows the time-space of the formula and makes it clearer to interpret (unambiguous). We first introduce the concept of *non-trivial formula decomposition*, and then we define the unambiguous predicate for a formula ϕ.

Definition 6 Non-trivial disjunctive formula decomposition. *The set of* non-trivial disjunctive decomposition *of a formula ϕ, written* $\mathsf{NTD}(\phi)$*, is defined as the set of non*

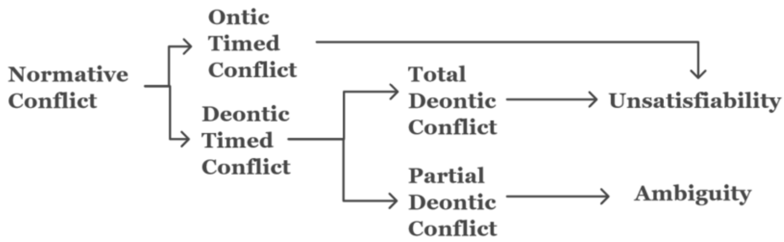

Figure 1. Overview: relation between normative conflicts and satisfiability

trivial[3] *pairs of formulas* (ϕ_i, ϕ_j) *such that* $\phi_i \neq \phi$ *and* $\phi_j \neq \phi$. *The set is defined as follows:* $\mathsf{NTD}(\phi) = \{(\phi_1, \phi_2) \mid \phi_1 \sqcup \phi_2 \equiv \phi$ *and* $\phi_1 \neq \phi$ *and* $\phi_j \neq \phi\}$.

Definition 7 Unambiguous satisfiability and max bad sub-formula. *A formula* ϕ *is* unambiguously satisfiable *(*$\mathsf{U-sat}$*) if and only if all pairs in the non-trivial decompositions of the formula are sat:* ϕ *is* $\mathsf{U-sat} \iff \forall(\phi_i, \phi_j) \in \mathsf{NTD}(\phi)$: ϕ_i *is* sat *and* ϕ_j *is* sat
We say that ϕ_b *is the maximal bad sub-formula of* ϕ *written* $\mathsf{Mbad}(\phi)$ *iff it contains all unsatisfiable sub-formulas of* ϕ.

Example 1. *Consider the formula:* $\phi = O^{\{[0,9]\}}(a) \sqcap F^{\{[6,9]\}}(a)$. ϕ *is satisfiable but contains a deontic timed conflict. Following definition* $\mathsf{U-sat}$, *we have:*
$\phi \equiv (O^{\{[0,7]\}}(a) \sqcap F^{\{[6,9]\}}(a)) \sqcup (O^{\{[7,9]\}}(a) \sqcap F^{\{[6,9]\}}(a))$. *The right sub-formula is unsat. There are multiple decompositions for which one sub-formula is unsat. Among them, the maximal unsat sub-formula is:* $\mathsf{Mbad}(\phi) = O^{\{[6,9]\}}(a) \sqcap F^{\{[6,9]\}}(a)$. *Thus the unambiguous rewriting for* ϕ *is* $(O^{\{[0,5]\}}(a) \sqcap F^{\{[6,9]\}}(a))$.

Lemma 4.3. *If* ϕ *contains a deontic time conflict then* ϕ *is unsat or* ϕ *is not* $\mathsf{U-sat}$.

Automatic partial conflict reparation We could randomly pick a decomposition of the formula and use the SMT solver to check the satisfiability of both sub-formulas and gradually eliminate the unsatisfiable parts. However, this naïve approach does not always converge, since we could have infinite intervals as constraints. We have established syntactic rules based on the arithmetic of sets of intervals to improve the approach.

5. Conclusion and Future work

In this paper, we studied the notion of normative conflicts in metric time settings. Using a minimal logic, we distinguished *ontic timed conflicts* from *deontic timed conflicts* as well as the notion of *total time deontic conflict* from *partial time deontic conflict*. We mapped total time conflict to unsatisfiability and partial conflict to *ambiguity*. We discussed repairing partial conflicts; in future work we will present an efficient algorithm to solve it.

The FMMTNL logic has limited expressiveness. We plan to *enrich the reasoning framework* to deal with timed operators such as sequence, reparation, and conditional

[3] $(\phi, O^{\emptyset}(a))$ is a trivial decomposition for ϕ.

norms as well as a model setting such as dependencies between actions to define new types of conflicts. Also, we will study the potential use of *defeasible semantics* for the resolution of total timed conflicts in the resulting enriched logic.

References

James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

Shaun Azzopardi, Gordon Pace, Fernando Schapachnik, and Gerardo Schneider. On the specification and monitoring of timed normative systems. In *International Conference on Runtime Verification*, pages 81–99. Springer, 2021.

John J Camilleri, Gabriele Paganelli, and Gerardo Schneider. A cnl for contract-oriented diagrams. In *Controlled Natural Language: 4th International Workshop, CNL 2014, Galway, Ireland, August 20-22, 2014. Proceedings 4*. Springer, 2014.

Silvano Colombo Tosatto, Guido Governatori, and Pierre Kelsen. Detecting deontic conflicts in dynamic settings. In *Deontic Logic and Normative Systems: 12th International Conference, DEON 2014, Ghent, Belgium, July 12-15, 2014. Proceedings 12*, pages 65–80. Springer, 2014.

Frank Dignum and Ruurd Kuiper. Obligations and dense time for specifying deadlines. In *Thirty-First Annual Hawaii International Conference on System Sciences, Kohala Coast, Hawaii, USA, January 6-9, 1998*, pages 186–195. IEEE Computer Society, 1998. . URL https://doi.org/10.1109/HICSS.1998.648312.

William M Farmer and Qian Hu. Fcl: A formal language for writing contracts. In *Quality Software Through Reuse and Integration*. Springer, 2016.

Stephen Fenech, Gordon J Pace, and Gerardo Schneider. Automatic conflict detection on contracts. In *International colloquium on theoretical aspects of computing*. Springer, 2009.

Guido Governatori and Antonino Rotolo. Justice delayed is justice denied: Logics for a temporal account of reparations and legal compliance. In *CLIMA'11*. Springer, 2011.

Guido Governatori, Joris Hulstijn, Régis Riveret, and Antonino Rotolo. Characterising deadlines in temporal modal defeasible logic. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 2007.

Tom Hvitved, Felix Klaedtke, and Eugen Zalinescu. A trace-based model for multiparty contracts. *J. Log. Algebraic Methods Program.*, 81(2), 2012.

Karam Younes Kharraz, Martin Leucker, and Gerardo Schneider. Timed dyadic deontic logic. In *Legal Knowledge and Information Systems*. IOS Press, 2021.

Giovanni Sartor. Normative conflicts in legal reasoning. *Artif. Intell. Law*, 1(2-3):209–235, 1992.

Luciano H Tamargo, Diego C Martinez, Antonino Rotolo, and Guido Governatori. An axiomatic characterization of temporalised belief revision in the law. *Artificial Intelligence and Law*, 27:347–367, 2019.