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

doi:10.3233/FAIA241251

Conflict Analysis for Timed Contract Automata

Shaun AZZOPARDI [a] and Gordon J. PACE [b]
[a] *Independent Researcher*
[b] *University of Malta, Malta*

Abstract. One can find various temporal deontic logics in literature, most focusing on discrete time. The literature on real-time constraints and deontic norms is much sparser. Thus, many analysis techniques which have been developed for deontic logics have not been considered for continuous time. In this paper we focus on the notion of conflict analysis which has been extensively studied for discrete time deontic logics. We present a sound, but not complete algorithm for detecting conflicts in timed contract automata and prove the correctness of the algorithm, illustrating the analysis on a case study.

Keywords. Deontic Logic, Timed Contract Automata, Conflict Analysis

1. Introduction

In the literature, one can find various variations of temporal deontic logics [9], however, much of it is limited to discrete model of time. Although one may argue that, with sufficiently fine temporal granularity, discrete time logics suffice, specifications and legal clauses may include real-time notions, which may not be easily mappable to a discrete setting. We find some deontic logics able to deal with real-time in literature. Governatori et al. have presented various partial formalisations of normative specifications with time e.g. [9,10,11], using *defeasible logic*. C-O Diagrams [5] use a formal visual representation of normative systems and are given a semantics using timed automata semantics. Themulus [8] is a calculus-based approach using a bisimulation-based approach to enable contract comparison. Kharraz et al. [12] developed a timed version of a dyadic deontic logic. More recently, timed contract automata [4] were presented as a real-time extension of contract automata [2] adding clocks, inspired by *timed automata*.

The concept of deontic conflicts has long been studied in the literature [6]. For instance, an agreement which (i) obliges a person who holds a resource to release it when another party requests it, and (ii) prohibits releasing the resource halfway through a high-priority transaction, would result in a conflict when a party is halfway through a transaction and another party requests it. The notion of conflict goes beyond prohibition vs. obligation, but also covers permissions [7], mutually exclusive actions [7,2] and general environmental constraints [13]. However, the notion of conflict in the context of real-time deontic logics has not, to the

best of our knowledge, been explored in the literature. Real-time introduces new challenges to conflict analysis. For instance, we can extend the example above to cover real time: (i) a person who holds a resource is obliged to release it within 15 minutes of another party requesting it, and (ii) it is prohibited to release a resource halfway through a high-priority transaction. Note that, adding a third clause which states that (iii) high-priority transactions may not be started while a request is pending, the conflict would be resolved.

In this paper, we explore conflict analysis in the context of real-time deontic logics, focussing on timed contract automata, where the main issue with conflict analysis lies in the presence of norms that outlive the explicit automaton state they are triggered from. Our algorithm is proved to be sound, but is not complete.[1]

2. Normative Conflicts

Due to space limitations, we refer the reader to [4] for the semantics of timed contract automata. We start by characterising our notion of conflict, to enable reasoning about the correctness of the algorithm discovering them. We start with the notion of a conflict at a particular point in time.

Definition 1. *A set of norms N is said to conflict at a clock valuation v, written $\maltese (N, v)$, when there is either an obligation or a permission, and a prohibition on the same action that are both active at v i.e. there exists $F_{\tau'}(p : a) \in N$ and $O_\tau(p : a)$ or $P_\tau(p : a)$ in N such that $\tau(v) \wedge \tau'(v)$.*

This notion of conflicts corresponds to the notion of conflicts defined in the discrete time setting for contract automata [2]. Using the definition above, we can talk about the notion of conflicts arising in a timed contract automaton.

Definition 2. *A timed contract automaton M is said to be* conflict-free*, written conflict-free(M), if all timed traces will put the automaton into a configuration with no conflict: $conf_0 \stackrel{ts}{\Longrightarrow} (q,\ v,\ P,\ E) \implies \neg \maltese (P \cup E, v)$.*

It is worth noting that we identify as a conflict any point in time in which opposing norms are in force. For example, a situation where a party is obliged to perform an action between time 0 and 10, but prohibited from doing the same action between time 0 and 5 is a conflict. A more lenient definition of conflict would have allowed this, since the obligation is still satisfiable. Both are useful notions. Here we take the former view, allowing us to capture and provide to the user more useful information about how we structure their behaviour. We leave the alternative interpretation as future work.

3. From Persistent Norms to Ephemeral Norms

The difficulty in finding conflicts primarily lies in persistent norms. If the timed contract automata only had ephemeral norms, it would be a matter of checking

[1] For the sake of conciseness, we do not cover timeouts in this paper, but we believe that the approach we have taken can be readily extended to deal with them.

for conflicts locally in the states. Persistent norms, however, outlive the state they lie in, and the key to the conflict analysis analysis we present is to transform persistent norms into ephemeral ones. In the process we also complete the automaton (such that the do-nothing semantic rule is never triggered). We first define some useful functions.

Our reduction of persistent norms to ephemeral norms is exponential in the number of norms used. This is since we consider (almost) all the possible subsets of a set of norms that can be satisfied at some point in time. Given such a subset, we replicate transitions in the original automaton using the conjunction of their condition with the timing conditions during which the norms may be satisfied, and only of those norms.

Abstracting active norms: Recall the *active* function, which given a set of norms N, an action $p : a$, and a timed valuation v returns N without the norms satisfied by $(p : a, v)$. We abstract this with $active_\alpha : \mathbb{N} \times \mathbb{P} \times \mathbb{A} \to 2^{\mathbb{N}}$, which ignores timing constraints. Instead of returning one set, it returns a set of subsets of N possibly satisfied when the action $p : a$ occurs (at any point in time). We later show that for every timed valuation v, this abstract set contains the concrete set of active norms. Given an action $p : a$ and a set of norms Ns, we define $active_\alpha(N, p : a) = \{N' \subseteq N \mid \forall O_\tau(p' : a') \in N \cdot p' : a' \neq p : a \implies O_\tau(p' : a') \in N'\}$.

Timing conditions: To characterise the timing condition required to discharge atomic norms, we define the function tc mapping $\mathcal{O}_\tau(p : a)$ to τ and permissions and prohibitions to $\lambda v.v > \max(\tau)$. We overload this for sets of norms.

We will use tc to identify when a set in $active_\alpha$ is the only set of norms satisfied at some time point. Essentially, given a choice of $N' \in active_\alpha(N, p : a)$, we want to be able to express a timing constraint T that is only true when there is a v s.t. $N' = active(N, (p : a, v))$. For ease of exposition, we define T for a set of norms that are satisfied (N_{sat}) and not satisfied ($N_{\neg sat}$).

Given two norm sets N_{sat} and $N_{\neg sat}$ we define the timing condition required for all norms in N_{sat} to be satisfied and for all norms $N_{\neg sat}$ not to be satisfied: $\mathrm{T}(p : a, N_{sat}, N_{\neg sat})$, defined as $tc(N_{sat}) \wedge \neg \bigvee tc(\mathcal{P}(N_{\neg sat}) \cup \mathcal{F}(N_{\neg sat}) \cup \{O_\tau(p : a) \in N_{\neg sat}\})$. Note that we treat obligations differently from the other types of norms, namely we ignore obligations not over $p : a$. These are immediately not satisfied by $p : a$ occurring, and not the action they predicate over. This definition does not deal with such obligations being in N_{sat}, but our use of T will never include these in N_{sat}.

The timed semantics uses two different kinds of transitions: explicit transitions and implicit transitions which are treated differently in our construction.

Translation: From an automaton $M = \langle Q, q_0, \to, pers, eph\rangle$ our translation creates automaton $M^+ = \langle Q^+, q_0^+, \to^+, \emptyset, eph^+\rangle$, defined below:

1. States, $Q^+ \in Q \times 2^{\mathbb{N}} \times 2^{\mathbb{N}}$, keep track of the active norms s.t. $(q, E, P) \in Q^+$ iff $q \in Q$, $E \subseteq eph(q)$, and $P \subseteq \{N \in pers(q') | q' \in Q\}$.
2. $q_0^+ \stackrel{df}{=} (q_0, eph(q_0), pers(q_0))$,
3. $eph^+((q, E, P)) = E \cup P$,
4. $\to^+$ is defined as the smallest relation constructed by the following rules:

(a) **Explicit transitions**: Given a state (q, E, P), and a transition from q in the original automaton, we consider all possible subsets of the originally persistent norms that can be satisfied by this transition, along with their conditions for satisfaction. Based on these, we create a new transition, leaving only the norms left to be satisfied, adding norms relevant to the new state, and adding the timing conditions for satisfaction of the removed norms to the guard.

$$\frac{q \xrightarrow{p:a|\tau\mapsto\rho} q' \qquad P' \in active_\alpha(P, p:a)}{(q,E,P) \xrightarrow{p:a|\tau\wedge \mathrm{T}(p:a,P\setminus P',P')\mapsto\rho} (q', eph(q'), P' \cup pers(q'))}$$

(b) **Implicit transitions**: For each state q and action $p : a$, construct a condition that captures when no corresponding transition is triggered. Further consider all possible subsets of the ephemeral and originally persistent norms that can be satisfied by performing $p : a$. Based on these, we create a corresponding new self-loop transition, with a guard capturing the implicit transition triggering and the satisfaction of the guessed norms.

$$\frac{\tau' = \neg \bigvee\{\tau \mid q \xrightarrow{p:a|\tau\mapsto\rho} q'\} \qquad E' \in active_\alpha(E, p:a) \qquad P' \in active_\alpha(P, p:a)}{(q,E,P) \xrightarrow{p:a|\tau'\wedge\tau_{to}\wedge \mathrm{T}(p:a,E\setminus E',E')\wedge \mathrm{T}(p:a,P\setminus P',P')\mapsto\rho_{id}} (q, E', P')}$$

Note how this reduction maintains the determinism of the original automaton, given we just refine the transitions of M further and keep track of satisfaction and activation of norms.

Theorem 1. *For all timed traces ts, there are sets of timed norms P and E such that $conf_0 \overset{ts}{\underset{M}{\Longrightarrow}} (q, v, P, E)$ iff $conf_0^+ \overset{ts}{\underset{M^+}{\Longrightarrow}} ((q, E, P), v, \emptyset, E \cup P)$ (where $conf_0^+ \overset{df}{=} (q_0^+, v_0, \emptyset, eph(q_0^+))$.*

From this theorem, we can immediately conclude that the two automata are equivalent with respect to violation and conflict-freedom.

Corollary 1. *A timed contract automaton is (i) violated if and only if its flattening is violated; and (ii) conflict-free if and only if its flattening is conflict-free.*

Complexity Given the definition and use of $active_\alpha$ the construction is exponential in the maximum sum, given a state in M, of: (1) the number of persistent obligations over the same action and with different timing predicates; and (2) the number of permissions and prohibitions.

4. Conflict Analysis

Since the reduction to ephemeral norms maintains the same semantics as the original automaton, we can use it to find conflicts in M.

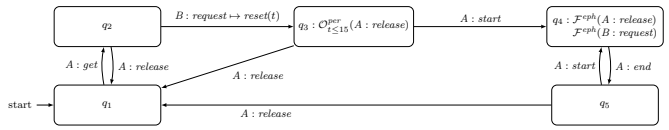
Theorem 2. *Given a timed automata M, if there is no local conflict in M^+, then M is conflict-free: If for all states (q, E, P) in Q_{M^+} $\neg(\textit{local-conflict}(E \cup P))$ holds, then conflict-free(M).*

This approach is sound but not complete, since some conflicting states may not be reachable. We can make the reduction finer by pruning away transitions with unsatisfiable conditions, however this is not sufficient for completeness. For example, consider a conflicting state that has an obligation that is required to be satisfied before the conflicting norms are activated, while satisfying the obligation implies leaving the state. Moreover, the state may be only possibly visited after at least one of the timing conditions of the conflicting norms no longer can hold.

We believe a sound and complete conflict analysis may be possible, through further transforming the ephemeral flattening into a timed safety automaton with special sink states denoting conflicts. Then conflict analysis can be reduced to reachability of timed automata, which is PSPACE-complete [3]. Given we are already paying an exponential cost to reduce persistent norms into ephemeral norms, here we prefer to rely on the sound algorithm given it can capture all the possible conflicts.

5. Case Study

To illustrate our approach we consider the case study discussed in Section 1. Consider the following three clauses: (i) Party A is obliged to release a resource within 15 minutes of party B requesting it (if they hold it); (ii) It is forbidden for party B to request a resource during a high-priority transaction; (iii) It is prohibited for party A to release a resource during a high-priority transaction. The figure below illustrates part of the corresponding automaton:



Due to the lack of space, we do not show the ephemeral flattening, but note that the flattening will only affect state q_4 and q_5, adding the persistent obligation in q_3 to states q_4 and q_5. The only potential for conflicts is in q_4 containing both the obligation to and prohibition from releasing the resource. Adding a prohibition on party A from starting a high-priority transaction while a request is in place (i.e. in state q_3) would resolve this conflict.

6. Conclusions

In this paper, we have presented an algorithm for the discovery of conflicts in clock reset-free timed contract automata. Detailed proof of its correctness can be found in [1]. The algorithm is sound, but not complete. Real-time logics push various analyses beyond the reach of the computable and we intend to explore further timed contract automata, investigating both algorithms to perform certain analyses, and proving impossibility results.

References

[1] Shaun Azzopardi and Gordon J. Pace. Sound conflict analaysis for timed contract automata. *CoRR*, abs/2410.12585, 2024.

[2] Shaun Azzopardi, Gordon J. Pace, Fernando Schapachnik, and Gerardo Schneider. Contract automata - an operational view of contracts between interactive parties. *Artif. Intell. Law*, 24(3):203–243, 2016.

[3] Patricia Bouyer and François Laroussinie. Model checking timed automata. *Modeling and Verification of Real-Time Systems: Formalisms and Software Tools*, pages 111–140, 2010.

[4] Stefan Chircop, Gordon J. Pace, and Gerardo Schneider. An automata-based formalism for normative documents with real-time. In Enrico Francesconi, Georg Borges, and Christoph Sorge, editors, *Legal Knowledge and Information Systems - JURIX 2022: The Thirty-fifth Annual Conference, Saarbrücken, Germany, 14-16 December 2022*, volume 362 of *Frontiers in Artificial Intelligence and Applications*, pages 158–163. IOS Press, 2022.

[5] Gregorio Díaz, María-Emilia Cambronero, Enrique Martínez, and Gerardo Schneider. Specification and verification of normative texts using C-O Diagrams. *Transactions on Software Engineering*, 40(8):795–817, 2014.

[6] Stephen Fenech, Gordon J. Pace, and Gerardo Schneider. Automatic Conflict Detection on Contracts. In *ICTAC'09*, volume 5684 of *LNCS*, pages 200–214. Springer, 2009.

[7] Stephen Fenech, Gordon J. Pace, and Gerardo Schneider. CLAN: A tool for contract analysis and conflict discovery. In *ATVA'09*, volume 5799 of *LNCS*, pages 90–96. Springer, 2009.

[8] Alberto García, María-Emilia Cambronero, Christian Colombo, Luis Llana, and Gordon J. Pace. Themulus: A timed contract-calculus. In *MODELSWARD'20*, pages 193–204. SciTePress, 2020.

[9] Guido Governatori, Joris Hulstijn, Régis Riveret, and Antonino Rotolo. Characterising deadlines in temporal modal defeasible logic. In *AI'07*, pages 486–496, 2007.

[10] Guido Governatori and Antonino Rotolo. Justice delayed is justice denied: Logics for a temporal account of reparations and legal compliance. In *CLIMA XII*, pages 364–382, 2011.

[11] Guido Governatori, Antonino Rotolo, and Giovanni Sartor. Temporalised normative positions in defeasible logic. In *ICAIL'05*, pages 25–34, 2005.

[12] Karam Younes Kharraz, Martin Leucker, and Gerardo Schneider. Timed dyadic deontic logic. In *The 34th International Conference on Legal Knowledge and Information Systems (JURIX'21)*, volume 346 of *Frontiers in Artificial Intelligence and Applications*, pages 197–204. IOS Press, 2021.

[13] Gordon J. Pace. A general theory of contract conflicts with environmental constraints. In Serena Villata, Jakub Harasta, and Petr Kremen, editors, *Legal Knowledge and Information Systems - JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, volume 334 of *Frontiers in Artificial Intelligence and Applications*, pages 83–92. IOS Press, 2020.