A General Theory of Contract Conflicts with Environmental Constraints

Gordon J. PACE [a,1]
[a] *University of Malta, Msida, Malta*

Abstract. One advantage of using formal deontic logic to represent and reason about normative texts is that one can analyse such texts in a precise and incontrovertible manner. Conflict analysis is one such analysis technique — assessing whether a number of contracts, or more generally normative texts, are internally consistent, in that they may not lead to a situation in which active norms conflict or even contradict each other. In this paper we extend existing techniques from the literature to address conflicts in the context of environmental constraints on actions regulated by the contract, and which the parties involved can carry out. The approach is logic-agnostic and we show how it can be applied to a service provision contract written in $\mathcal{CL}$.

Keywords. deontic logic, contract conflicts

1. Introduction

Lessig's *code is law* dictum is based on the notion that there are different means of restricting or regulating behaviour. Lessig identifies different forms of such means, including legal, and architectural (or by physical design). For instance, by the very nature of the real world, physical laws cannot be violated, and thus restrict our behaviour no matter the legal constraints one may have. To use an example used by Lessig, until the invention of human means of flight, a law giving a person rights to his or her land and everything above it (and below — the principle of *'cuius est solum eius est usque ad coelum et ad infernos'*, or *'the owner of the land all the way up to heaven, and the way down to hell'*) was no different to a law which gives rights only over the first 100 metres above the owned land. Conversely, obliging a person to be in two geographically distant points $O(inRome) \wedge O(inGlasgow)$ is unsatisfiable due to the physical nature of the two actions or states of affairs. Such constraints implicitly limiting one's behaviour have (arguably) become more common with the rise of computer code — hence Lessig's *'code is law'*. Computer code which prohibits downloading a file unless one is logged in, and logging in requires to have registered before, which in turn requires to have provided personal information, means that an agreement which prohibits you from ever giving personal details is implicitly incompatible with one which obliges you to download a particular file. In a free-for-all world, the two agreements are compatible, but not in the world where computer code is regulating our interaction with the server.

The notion of conflicts in deontic logics [8] has been investigated before in order to identify situations which may arise from a formula in which at least one of the parties

[1] Corresponding author; E-mail: gordon.pace@um.edu.mt

has conflicting norms to satisfy e.g. being both obliged and prohibited from performing a particular action. Much of the work in the literature assume that there are no restrictions on the actions or states being regulated, although some approaches allow for the consideration of mutually exclusive actions. Such a consideration allows us to identify the conflict in $O(inRome) \wedge O(inGlasgow)$ if we know that the two actions are mutually exclusive. However, limiting oneself to mutually exclusive actions is not strong enough to deal with more complex environmental constraints or Lessig's code. For instance, in the file downloading example, the constraint is more sophisticated than a mere mutual exclusion of actions — with a four-fold temporal dependancy: *download* cannot happen before *login*, which in turn cannot happen before *registration* which cannot happen before *sendPersonalDetails*.

In this paper we investigate the interaction between environmental (or factual) constraints and contract conflicts. We define a general framework for temporal deontic logics and general action constraints, in which we characterise the notion of a conflict in a formula in the deontic logic under particular constraints. In order to show the use of this general framework, we show how it can be applied to the contract language $\mathcal{CL}$ [1] and use it to analyse an internet service provider contract for conflicts under temporal constraints arising from the underlying computer system.

2. Related Work

Detection of conflicts between normative documents, as a first step towards eliminating them if possible (e.g. when a contract is still being drafted) or to resolve or reconcile them when not (e.g. when such a conflict arises from existing contracts during a business acquisition), has long been considered an important challenge [19]. With normative documents expressed in the form of a deontic logic one has the opportunity of formally characterising the class of such conflicts and derive algorithms to extract them automatically, and one finds various such work for particular contract languages.

One of the first formal and computational characterisations of deontic conflicts and algorithms to detect them was for the contract language $\mathcal{CL}$ [1]. Fenech et al. [9] characterised the notion of conflicts in $\mathcal{CL}$ formulae through the use of a trace semantics of the deontic logic, with the analysis integrated in the conflict detection CLAN tool [10]. The conflict analysis was used for other deontic logics such as C-O diagrams [16]. Other approaches to conflict discovery taking a definitional approach to characterising conflicts include [14] using defeasible logic, [21] which takes a unification-based approach, [20] which addresses conflicts in dynamic settings [13] which addresses conflict by taking by devising preferential rules, and [12] which defines a notion of *coherence*.

An alternative axiomatic approach to conflict discovery was used to identify conflicts in contract automata [6]. Unlike the former work, the notion of conflicts is captured in a number of axioms which characterise how conflicts arise and under which conditions they are maintained, but similar to the work in $\mathcal{CL}$, the approach also took an operational view of the deontic formulae inherent in the automata used.

Although the approaches mentioned above focus on formal representations of normative documents, there is work on using them for natural language texts, from using controlled natural languages e.g. see [3] to free text contracts e.g. see [2, 5, 4].

The approach we present in this paper takes a deontic logic-agnostic approach to characterising conflicts, but furthermore, we extend the analysis to take into account environmental constraints which may give rise to conflicts.

3. A General Model of Temporal Deontic Logics and Constraints

We assume that the normative texts will be with reference two parties $\mathbb{P} \overset{df}{=} \{1, 2\}$. We will use variables p, p' to range over $\mathbb{P}$, and will write $\overline{p}$ to refer to the party other than p.

Since we will be looking at action-based deontic logics, we will assume an alphabet Σ of possible actions, with variables a, a' ranging over this alphabet. In order to identify which party has attempted (or performed) an action, we will tag actions by the participating party: $\Sigma_{\mathbb{P}} \overset{df}{=} \{a_p \mid a \in \Sigma,\ p \in \mathbb{P}\}$. Furthermore, in order to enable multiple actions occurring simultaneously, we will look at actions sets ranging over the power set of the alphabet $2^{\Sigma_{\mathbb{P}}}$ which we will refer to as $\mathbb{\Sigma}_{\mathbb{P}}$, with variables A and A' ranging over this type. Finally, we will also need to refer to finite sequences of action sets $\mathbb{\Sigma}_{\mathbb{P}}^*$, using variables $\overline{A}$ and $\overline{A}'$ to range over them.

3.1. Deontic logics

Since our intention is to develop a general framework for action-based deontic logics in general, rather than for a particular one, we distill requirements for conflict analysis to a number of basic predicates and relations over the underlying logic we wish to analyse.

A deontic logic can be characterised by a set of well-formed formulae *DeonticFormula* in the logic, using variables ψ, ψ' to range over these well-formed formulae. We will be dealing with deontic logics which include a notion of discrete time, and will assume a derivative operational relation [7] expressing how a formula evolves when a set of actions is performed, writing $\psi \xrightarrow{A} \psi'$ to denote that upon receiving set of actions A, the residual formula of ψ (the new formula encoding the state of the contract) is ψ'. For instance, in the contract logic $\mathcal{CL}$ [1], one may write the contract $[a]\mathcal{O}(b)$ to indicate that *'if a is initially performed, then on obligation to perform b is enacted, and if not, no residual contract remains.'* The derivative relation would include $[a]\mathcal{O}(b) \xrightarrow{\{a\}} \mathcal{O}(b)$, $[a]\mathcal{O}(b) \xrightarrow{\{a,c\}} \mathcal{O}(b)$ and $[a]\mathcal{O}(b) \xrightarrow{\{c\}} \top$. We will write $\psi \overset{\overline{A}}{\Rightarrow} \psi$ to indicate the transitive closure of single step derivatives over action set trace $\overline{A}$.

At the core of all deontic logics, we require the underlying normative literal clauses which identify immediate norms in force and which the logic can handle. For instance, standard deontic logic [11] and the contract language $\mathcal{CL}$ would include the notion of obligation to perform action a: $\mathcal{O}(a)$, prohibition to perform action a: $\mathcal{F}(a)$ and permission to do so: $\mathcal{P}(a)$. In contract automata [6], these literals are parametrised by the party to whom they apply: $\mathcal{O}_p(a)$, $\mathcal{F}_p(a)$ and $\mathcal{P}_p(a)$. In all these cases, the norm can also be applied to absence of an action e.g. in contract automata one can have $\mathcal{O}_p(!a)$. We assume a set of norm literals NormLiteral, and use variable ∂ to range over this set. Furthermore, we will assume the notion of the norm opposing ∂, written $!\partial$, with the operator being a partial function from NormLiteral to NormLiteral such that, when well defined, $!!\partial = \partial$. For instance, many deontic logics would have $!\mathcal{P}(a) = \mathcal{F}(a)$ and $!\mathcal{O}(a) = \mathcal{P}(!a)$. Given a set of norm literals in force $\mathcal{D} \subseteq$ NormLiteral, we will assume that there is a predicate $vio_A(\mathcal{D})$ which holds if action set A is in violation of literal set $\mathcal{D}$[2].

[2] It is worth noting that this cannot always be reduced to a relation on single norm literals. For instance, in interactive systems, the counter-party of a permission must provide an action set which contains *all* permitted actions, not just an action set for each permitted actions. See [6] for more details.

For a deontic logic we will assume that we have a way to extract the set of normative literals $\mathcal{D} \subseteq \text{NormLiteral}$ that are in force upon enacting that deontic formula, through the function: $\text{norms}_0 \in \mathit{DeonticFormula} \rightarrow 2^{\text{NormLiteral}}$. For example, in $\mathcal{CL}$, the formula $O(a) \wedge [b]\mathcal{F}(c)$ only enforces an obligation to perform a now, with prohibition to perform c only to be enacted if b is initially performed. Thus, we would expect that $\text{norms}_0(O(a) \wedge [b]\mathcal{F}(c)) = \{O(a)\}$.

Definition 1. *A temporal deontic logic over alphabet Σ is characterised as a tuple $\langle \Sigma, \mathit{NormLiteral}, \mathit{DeonticFormula}, \mathit{norms}_0, \mathit{vio}, \mapsto \rangle$, where (i) NormLiteral are the basic underlying norm literals the logic can express; (ii) DeonticFormula is the set of well-formed formulae in the logic; (iii) $\mathit{norms}_0 \in \mathit{DeonticFormula} \rightarrow 2^{\mathit{NormLiteral}}$ is a function giving the norms in immediate effect; (iv) $\mathit{vio} \in \Sigma_{\mathbb{P}} \times 2^{\mathit{NormLiteral}} \rightarrow \mathbb{B}$ is the violation predicate which formalises when an action set violates a set of deontic literals; and (v) $\mapsto \in \mathit{DeonticFormula} \times \Sigma_{\mathbb{P}} \rightarrow \mathit{DeonticFormula}$ is the derivative function expressing how the logic evolves over occurrence of actions.*

In the rest of the paper, we overload the violation relation to well-formed formulae: $\mathit{vio}_A(\psi) \overset{df}{=} \mathit{vio}_A(\mathit{norms}_0(\psi))$.

Many of the temporal deontic logics in the literature have been given such an operational semantics. For instance, the contract logic $\mathcal{CL}$ [1] was given semantics in this form in [9]. $\mathcal{CL}$ is given a trace semantics from which one can easily calculate the derivative function. The trace semantics also carry information as to which norms are in force, which provides for a definition of the norms_0 function. Finally, the violation predicate corresponds to action sets which would reduce the contract formula to $\bot$ (the implicitly violated contract in $\mathcal{CL}$).

Similarly, contract automata [6] formalise contracts between interacting parties as deterministic automata with (i) transitions labelled by sets of actions; and (ii) states tagged by a set of norm literals which are in force when in that state. There is a direct correspondence between the set of *DeonticFormula* with the states in the contract automaton, the derivative function with the transition relation of the automata and set of norms in immediate effect corresponding to the norms in the state of the automaton one is in. Contract automata already provide a violation predicate which assesses whether an action set is in violation of the norm literals in the current state, which can be used directly for the *vio* predicate.

In order to enable a complete axiomatisation of conflicts, we have to take into consideration the fact that some norm literals are related together. For instance, in interacting system agreements, an obligation on one party to perform an action subsumes permission to perform that action i.e. the other party is required to provide the necessary handshake to allow the action to happen [18]. In order to characterise such relations between norm literals, we will assume a strictness relation between sets of normative literals $\sqsubseteq$, such that $\mathcal{D} \sqsubseteq \mathcal{D}'$ holds if and only if $\mathcal{D}'$ is at least as strict as $\mathcal{D}$ i.e. would lead to at least as many violations. We assume a number of properties of the strictness relation:

Assumption 1. *(i) The strictness relation $\sqsubseteq$ is a partial order with $\emptyset$ being a minimum; (ii) If $\mathcal{D}' \sqsubseteq \mathcal{D}''$, then for any $\mathcal{D}$, $\mathcal{D} \cup \mathcal{D}' \sqsubseteq \mathcal{D} \cup \mathcal{D}''$.*

It is worth noting that from these assumed properties, it immediately follows that adding extra clauses can only make a contract stricter: $\mathcal{D} \sqsubseteq \mathcal{D} \cup \mathcal{D}'$.

3.2. Action predicates

In order to define constraints on the context, which limit what sets of actions are possible, we will use a boolean expressions over actions — such that action set A is possible if and only if the boolean expression holds when variables in A are instantiated to true, and all others to false. For example, to express the constraint that actions a and b are mutually exclusive, we would write this as $\neg(a \wedge b)$. Similarly, if we want to express the constraint that action a cannot appear unless b also appears, we would write $\neg b \implies \neg a$. As a final example, we can express that actions a, b and c cannot all appear together as $\neg(a \wedge b \wedge c)$.

Definition 2. *An* action constraint $\alpha \in \mathit{ActionConstraint}$ *is a boolean expression over actions defined using the following syntax:* $\alpha ::= \bot \mid \Sigma_{\mathbb{P}} \mid \neg\alpha \mid \alpha \vee \alpha$. *An action constraint corresponds to a collection of action sets defined as follows*[3]*:*

$$\begin{aligned}
[\![\bot]\!] &\overset{df}{=} \emptyset \\
[\![a]\!] &\overset{df}{=} \{A \mid A \subseteq \Sigma_{\mathbb{P}} \wedge a \in A\} \\
[\![\neg\alpha]\!] &\overset{df}{=} [\![\alpha]\!]^c \\
[\![\alpha \vee \alpha']\!] &\overset{df}{=} [\![\alpha]\!] \cup [\![\alpha']\!]
\end{aligned}$$

We will define other standard boolean operators in the usual manner: $\alpha \wedge \alpha' \overset{df}{=} \neg(\neg\alpha \vee \neg\alpha')$, $\alpha \implies \alpha' \overset{df}{=} \neg\alpha \vee \alpha'$ *and* $\alpha \iff \alpha' \overset{df}{=} (\alpha \implies \alpha') \wedge (\alpha' \implies \alpha)$.

We will overload the violation predicate, writing $\mathit{vio}_\alpha(\mathcal{D})$ *to indicate that all interpretations satisfying* α *violate* $\mathcal{D}$: $\forall A \in [\![\alpha]\!] \cdot \mathit{vio}_A(\mathcal{D})$.

An action constraint α *is said to be stronger than another action constraint* α', *written* $\alpha \vdash \alpha'$ *if and only if* $[\![\alpha]\!] \subseteq [\![\alpha']\!]$.

Since action constraints can vary over time (e.g. once locked, the door cannot be unlocked without the keycard being swiped), we extend action constraints temporarily using an approach similar to the way we expressed temporal deontic logics i.e. a transition system made up of well-formed formulae in *TemporalActionConstraint*, such that for a formula $\tau \in \mathit{TemporalActionConstraint}$, $\mathrm{constraint}_0(\tau)$ gives the action constraint initially in force for formula τ, and $\mapsto$ is the temporal derivative function for the transition system.

Definition 3. *A* temporal action constraint language *over alphabet* Σ *is characterised as a tuple* $\langle \Sigma, \mathit{TemporalActionConstraint}, \mathit{constraint}_0, \mapsto \rangle$, *where (i) TemporalActionConstraint is the set of well-formed formulae in the temporal action constraint language; (ii)* $\mathit{constraint}_0 \in \mathit{TemporalActionConstraint} \to \mathit{ActionConstraint}$ *is a function giving the action constraint in effect at the beginning; (iii)* $\mapsto\ \in \mathit{TemporalActionConstraint} \times \Sigma_{\mathbb{P}} \to \mathit{TemporalActionConstraint}$ *is the derivative function expressing how formula in the language evolves over occurrence of actions.*

For instance, consider the use of the Safety Linear Time Logic (Safety LTL) [22] as a temporal action constraint language, which would allow us to express the constraint on environment behaviour that *once locked, the door cannot be unlocked without the key-*

[3] We write S^c to denote the complement of set S.

card being swiped as the formula *Door*, defined to be $G(lock \implies (\neg unlock\ W\ swipe))$[4]. Derivatives of Safety LTL formulae can be defined in a standard manner as used for runtime verification e.g. see [15] whilst the action constraint is taken to be the weakest formula which, if it holds, would reduce the LTL formula to *false*. For instance, the derivative of formula $G(lock \implies (\neg unlock\ W\ swipe))$ with respect to $\{lock\}$ would be $(\neg unlock\ U\ swipe) \wedge Door$, whilst $\text{constraint}_0(Door) = \neg(lock \wedge unlock \wedge \neg swipe)$ indicating that the door is not allowed to be locked and unlocked immediately.

As in the case of deontic logic formulae we showed earlier, we write $\tau \overset{\overline{A}}{\Longrightarrow} \tau$ to indicate the transitive closure of single step derivatives over action set trace $\overline{A}$.

4. Deontic Conflicts

Much of the literature conflicts are formalised as a binary relation between contracts. For instance, in [6], we used a binary relation between norms $\psi \maltese \psi'$, defined to contain (i) conflicts between opposite norms, (ii) obligations to perform mutually exclusive actions, and (iii) defined to be closed under symmetry and increased strictness. This approach works well since the only context constraint is that of pairwise mutually exclusive actions. However, when we enrich the class of constraints which may be used, conflicts between pairs of norm clauses no longer suffices.

Consider, for example, an environmental constraint which ensures that the three actions a, b and c can never occur together. Now consider the normative clauses which place an obligation on party p to perform each action separately: $O_p(a)$, $O_p(b)$ and $O_p(c)$. No two of these three clauses conflict with each other under the environmental constraint, but the three together are in conflict since they clearly cannot be satisfied together.

In order to deal with conflicts under such an enriched class of environmental constraints have to talk about conflicts over a *set* of normative clauses rather than limiting it to two clauses i.e. deducing that under the context constraint mentioned above, the set of normative clauses $\{O_p(a),\ O_p(b),\ O_p(c)\}$ is in conflict.

4.1. Conflicts in norm literal sets

We can now define the notion of conflict within a set of normative clauses as a predicate over sets of norm literals. Such a set of norms is in conflict if (i) there are opposing norm literals or (ii) if the context constraints result in an unsatisfiable contract. In addition, conflicts are closed under (i) increased strictness of the norms; and (ii) strengthening of constraints. These four principles provide the

Definition 4. *A set of norm literals $\mathcal{D}$ is said to have an internal conflict under context constraint χ, written $\maltese_\chi(\mathcal{D})$, if it follows from the following axioms:*
Axiom 1: *Opposing literal norms conflict: $\maltese_{true}(\{\partial,\ !\partial\})$.*
Axiom 2: *Conflicts may arise from norms impossible to satisfy due to action constraints: if $vio_\chi(\mathcal{D})$ then $\maltese_\chi(\mathcal{D})$.*
Axiom 3: *Conflicts are closed under increased strictness: if $\maltese_\chi(\mathcal{D})$ and $\mathcal{D} \sqsubseteq \mathcal{D}'$, then $\maltese_\chi(\mathcal{D}')$.*
Axiom 4: *Conflicts are closed under constraint strengthening: if $\chi' \vdash \chi$ and $\maltese_\chi(\mathcal{D})$, then $\maltese_{\chi'}(\mathcal{D})$.* □

[4] In LTL, property $G\pi$ holds for a trace if and only if π holds for any suffix of the trace, $\pi\ W\ \pi'$ holds if and only if π will hold on every suffix of the trace until π' holds (although π' may never hold, in which case π must continue holding indefinitely) and $X\,\pi$ which holds if π holds if we ignore the first event in the trace.

Consider a deontic norm set with obligations on all three actions a, b and c: $\mathcal{D} = \{O_p(a),\ O_p(b),\ O_p(c)\}$ and the constraint: $\chi = a \land b \implies \neg c$. We can show that $vio_\chi(\mathcal{D})$ and thus, by Axiom 2 that $\maltese_\chi(\mathcal{D})$.

As another example, consider a norm set which includes an obligation and prohibition to perform the same action: $O_p(a)$ and $\mathcal{F}_p(a)$. Firstly note that in a deontic logic in which permission is weaker than obligation i.e. $\mathcal{P}_p(a) \sqsubseteq O_p(a)$, the contrapositive holds: $!O_p(a) \sqsubseteq !\mathcal{P}_p(a)$. Now, by Axiom 1, $\maltese_\chi(\{O_p(a),\ !O_p(a)\})$, from which (and the inequality just given) it follows that $\maltese_\chi(\{O_p(a),\ !\mathcal{P}_p(a)\})$ using Axiom 3. However, $\mathcal{F}_p(a)$ is equivalent to $!\mathcal{P}_p(a)$, from which we can conclude that: $\maltese_\chi(\{O_p(a),\ \mathcal{F}_p(a)\})$ which can be extended to any set containing these norm literals using Axiom 4 and Assumption 1 of the strictness relation: $\maltese_\chi(\{O_p(a),\ \mathcal{F}_p(a)\} \cup \mathcal{D})$.

4.2. Temporal conflicts

The notion of internal conflicts can be extended beyond sets of norm literals by ensuring conflicts never arise no matter what input is received.

Definition 5. *Given a temporal deontic logic formula $\psi \in \mathit{DeonticFormula}$ and temporal action constraint $\tau \in \mathit{TemporalActionConstraint}$, we say that ψ has a conflict under constraint τ, written $\maltese_\tau(\psi)$ if, for some action set trace, the immediate deontic norms of the derivative deontic formula are in conflict under the immediate derivative constraint:*

$$\maltese_\tau(\psi) \overset{df}{=} \exists \overline{A} \in \Sigma_{\mathbb{P}}^{*} \cdot \forall \psi' \in \mathit{DeonticFormula} \cdot \forall \tau' \in \mathit{TemporalActionConstraint} \cdot$$
$$\psi \overset{\overline{A}}{\Rightarrow} \psi' \land \tau \overset{\overline{A}}{\Rightarrow} \tau' \implies vio_{constraint_0(\tau')}(norms_0(\psi'))$$

□

Using this definition, for instance, we can discover that under an environmental constraint that says that after action a, b and c cannot occur together: $\tau = G(a \implies \neg X\,(b \land c))$, the $\mathcal{CL}$ formula $\psi = [a](O(b) \land O(c))$ is in conflict. Using the singleton trace $\overline{A} = \langle\{a\}\rangle$, we can show that using derivatives of $\mathcal{CL}$ and LTL: $\psi \overset{\overline{A}}{\Rightarrow} O(a) \land O(b)$ and $\tau \overset{\overline{A}}{\Rightarrow} \neg(b \land c) \land \tau$. Using the definitions of constraint$_0$ and norms$_0$, we can show that a conflict arises by proving that $vio_{\neg(b\land c)}(\{O(a),\ O(b)\})$ which follows from the previous definitions.

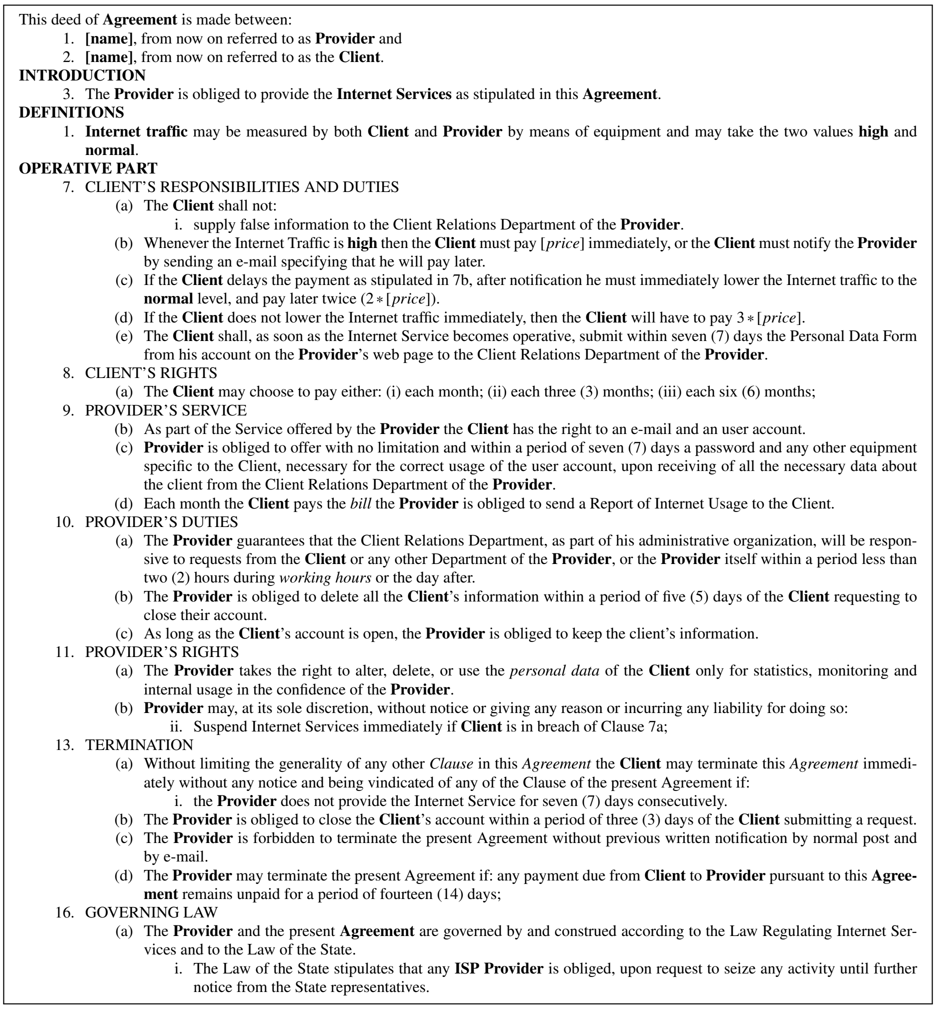
5. Use Case: An Internet Service Provider Contract in $\mathcal{CL}$

In order to investigate the use of the conflict analysis techniques we describes, we have use an instantiation of our contract conflict theory for $\mathcal{CL}$, and extended an Internet Service Provider contract from [17]. The contract between the service provider and the client, shown in Figure 1. In particular, note the client information deletion parts of the contract:

10(b) *The **Provider** is obliged to delete all the **Client**'s information within a period of five (5) days of the **Client** requesting to close their account.*

10(c) *As long as the **Client**'s account is open, the **Provider** is obliged to keep the client's information.*

13(b) *The **Provider** is obliged to close the **Client**'s account within a period of three (3) days of the **Client** submitting a request.*



This deed of **Agreement** is made between:

1. **[name]**, from now on referred to as **Provider** and
2. **[name]**, from now on referred to as the **Client**.

INTRODUCTION

3. The **Provider** is obliged to provide the **Internet Services** as stipulated in this **Agreement**.

DEFINITIONS

1. **Internet traffic** may be measured by both **Client** and **Provider** by means of equipment and may take the two values **high** and **normal**.

OPERATIVE PART

7. CLIENT'S RESPONSIBILITIES AND DUTIES
   (a) The **Client** shall not:
      i. supply false information to the Client Relations Department of the **Provider**.
   (b) Whenever the Internet Traffic is **high** then the **Client** must pay [*price*] immediately, or the **Client** must notify the **Provider** by sending an e-mail specifying that he will pay later.
   (c) If the **Client** delays the payment as stipulated in 7b, after notification he must immediately lower the Internet traffic to the **normal** level, and pay later twice ($2*[price]$).
   (d) If the **Client** does not lower the Internet traffic immediately, then the **Client** will have to pay $3*[price]$.
   (e) The **Client** shall, as soon as the Internet Service becomes operative, submit within seven (7) days the Personal Data Form from his account on the **Provider**'s web page to the Client Relations Department of the **Provider**.
8. CLIENT'S RIGHTS
   (a) The **Client** may choose to pay either: (i) each month; (ii) each three (3) months; (iii) each six (6) months;
9. PROVIDER'S SERVICE
   (b) As part of the Service offered by the **Provider** the **Client** has the right to an e-mail and an user account.
   (c) **Provider** is obliged to offer with no limitation and within a period of seven (7) days a password and any other equipment specific to the Client, necessary for the correct usage of the user account, upon receiving of all the necessary data about the client from the Client Relations Department of the **Provider**.
   (d) Each month the **Client** pays the *bill* the **Provider** is obliged to send a Report of Internet Usage to the Client.
10. PROVIDER'S DUTIES
   (a) The **Provider** guarantees that the Client Relations Department, as part of his administrative organization, will be responsive to requests from the **Client** or any other Department of the **Provider**, or the **Provider** itself within a period less than two (2) hours during *working hours* or the day after.
   (b) The **Provider** is obliged to delete all the **Client**'s information within a period of five (5) days of the **Client** requesting to close their account.
   (c) As long as the **Client**'s account is open, the **Provider** is obliged to keep the client's information.
11. PROVIDER'S RIGHTS
   (a) The **Provider** takes the right to alter, delete, or use the *personal data* of the **Client** only for statistics, monitoring and internal usage in the confidence of the **Provider**.
   (b) **Provider** may, at its sole discretion, without notice or giving any reason or incurring any liability for doing so:
      ii. Suspend Internet Services immediately if **Client** is in breach of Clause 7a;
13. TERMINATION
   (a) Without limiting the generality of any other *Clause* in this *Agreement* the **Client** may terminate this *Agreement* immediately without any notice and being vindicated of any of the Clause of the present Agreement if:
      i. the **Provider** does not provide the Internet Service for seven (7) days consecutively.
   (b) The **Provider** is obliged to close the **Client**'s account within a period of three (3) days of the **Client** submitting a request.
   (c) The **Provider** is forbidden to terminate the present Agreement without previous written notification by normal post and by e-mail.
   (d) The **Provider** may terminate the present Agreement if: any payment due from **Client** to **Provider** pursuant to this **Agreement** remains unpaid for a period of fourteen (14) days;
16. GOVERNING LAW
   (a) The **Provider** and the present **Agreement** are governed by and construed according to the Law Regulating Internet Services and to the Law of the State.
      i. The Law of the State stipulates that any **ISP Provider** is obliged, upon request to seize any activity until further notice from the State representatives.

Figure 1. Extracts from a contract between an internet service provider and their client

Using days for the unit of discrete time, we can encode the contract in $\mathcal{CL}$ e.g. clause 13(b) would be formulated[5] as: $[?^* \cdot requestTermination]O((1+?+?\cdot?)\cdot closeAccount)$. In addition, we have some logistical constraints arising from the technical setup of the service provider. Consider the following constraints about backups expressed in LTL:

$$\begin{aligned}
&\neg deleteClientInfoFromBackup\ W\ \neg deleteClientInfo && (1)\\
\wedge\ & deleteClientInfoFromBackup \implies X(\neg deleteClientInfoFromBackup\ W\ \neg deleteClientInfo) && (2)\\
\wedge\ & G(deleteClientInfoFromBackup \implies weeklyBackupInProgress) && (3)\\
\wedge\ & G(weeklyBackupInProgress \implies \textstyle\bigwedge_{1\leq i\leq 6} \neg X^i weeklyBackupInProgress) && (4)\\
\wedge\ & G(\neg(requestCloseAccount \wedge deleteClientInfo)) && (5)
\end{aligned}$$

[5] The formula says that any time the action guard (written in a regular expression like syntax, with ? meaning 'any action', star means repetition, + means choice and · indicates sequential composition) is satisfied i.e. a request for termination has just been received, the obligation to close the account within three time units is enacted (the action sequence of the obligation is also written in regular expression-like syntax.

Constraints (1) and (2) indicate that client information is not deleted from the backup unless it is deleted from the main database first. Constraint (3) further indicates that deletion from the backup can only occur when the weekly backup is taking place. Constraint (4) indicates that weekly backups never occur less than a week apart and finally, (5) indicates that due to demands for termination being processed in batch at the end of the day, a client's information is never deleted immediately upon a request to close the account.

Analysing the $\mathcal{CL}$ contract under these constraints will allow us to discover a conflict arising when the client requests to close their account triggering (i) an obligation to delete the client's information within five days including that kept in the backup (arising from clause 10(b)); (ii) an obligation to close the account within three days (from clause 13(b)); and (iii) an obligation to keep the data until the account is closed (from clause 10(c)). If the client request happed on the same day as backups are done, the constraints result in the backup not being deleted until at least 7 days have elapsed of the request, with the service provider thus not being able to comply to the contract's terms.

The implication of the discovery of this conflict is that either the contract is to be fixed to remove the conflict, or the internal systems must be changed in order to relax the constraints resulting in the conflict. The choice depends on the importance of the offending contract clauses and the possibility (or otherwise) of changing the constraints: Is it reasonable to start taking more frequent backups or to propagate deletions to the backups promptly, or is it simpler to extend the deletion period from five to 10 days?

6. Conclusions

In this paper we have presented a unified theory of temporal deontic contracts. Unlike previous work on the topic, we do not restrict ourselves to a particular deontic logic and, more importantly, allows taking into consideration temporal constraints on the actions taking place. We have illustrated the use of the theory by using its instantiation to the contract language $\mathcal{CL}$ and LTL for constraint expression in order to find conflicts in an Internet Service Provider contract taken from the literature.

We are currently looking at automated ways of automating the analysis and the modelling of different deontic and temporal logics in our model, including the use of symbolic model checking techniques in order to be able to explore contract sanity efficiently. Furthermore, our work can form the basis of conflict resolution, in order to refine or characterise text more effectively.

We also note that the theory can be extended to deal with other forms of contract analysis such as the detection of useless clauses — contract clauses which can be left out or simplified due to the fact that the environmental constraints will never have an effect e.g. a clause which states that *'the provider is obliged to either delete the data from the main database immediately, or to notify the user and delete it within 1 day'* can be simplified to *'the provider is obliged to notify the user and delete their data from the main database within 1 day'* due to the constraint that deletion cannot happen on the same day as a request for closing an account.

References

[1] A dynamic deontic logic for complex contracts. *The Journal of Logic and Algebraic Programming*, 81(4):458 – 490, 2012.

[2] João Paulo Aires, Daniele Pinheiro, Vera Strube de Lima, and Felipe Meneguzzi. Norm conflict identification in contracts. *Artif. Intell. Law*, 25(4):397–428, 2017.

[3] Krasimir Angelov, John J. Camilleri, and Gerardo Schneider. A framework for conflict analysis of normative texts written in controlled natural language. *J. Log. Algebraic Methods Program.*, 82(5-7):216–240, 2013.

[4] Shaun Azzopardi, Albert Gatt, and Gordon J. Pace. Integrating natural language and formal analysis for legal documents. In *Proceedings of the 10th Conference on Language Technologies and Digital Humanities 2016*, 2016.

[5] Shaun Azzopardi, Albert Gatt, and Gordon J. Pace. Reasoning about partial contracts. In *Legal Knowledge and Information Systems - JURIX 2016: The Twenty-Ninth Annual Conference*, pages 23–32, 2016.

[6] 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.

[7] Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, 1964.

[8] Xavier Parent Ron van der Meyden Leendert van der Torre Dov Gabbay, John Horty. *Handbook of Deontic Logic and Normative Systems*.

[9] Stephen Fenech, Gordon J. Pace, and Gerardo Schneider. Automatic conflict detection on contracts. In *Proceedings of Theoretical Aspects of Computing ICTAC*, 2009.

[10] Stephen Fenech, Gordon J. Pace, and Gerardo Schneider. CLAN: A tool for contract analysis and conflict discovery. In *Proceedings of Automated Technology for Verification and Analysis, 7th International Symposium, ATVA*, 2009.

[11] Georg Henrik Von Wright. Deontic Logic. *Mind*, 60(237):1–15, January 1951.

[12] Daniel Gorín, Sergio Mera, and Fernando Schapachnik. Model checking legal documents. In *The Twenty-Third Conference on Legal Knowledge and Information Systems (JURIX)*, volume 223 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2010.

[13] G. Governatori, F. Olivieri, M. Cristani, and S. Scannapieco. Revision of defeasible preferences. *International Journal of Approximate Reasoning*, 104:205–230, 2019.

[14] Guido Governatori and Robert Mullins. Deontic closure and conflict in legal reasoning. In *Legal Knowledge and Information Systems (JURIX)*, volume 322 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2019.

[15] Klaus Havelund and Grigore Rosu. Monitoring programs using rewriting. In *16th IEEE International Conference on Automated Software Engineering (ASE 2001), 26-29 November 2001, Coronado Island, San Diego, CA, USA*, pages 135–143. IEEE Computer Society, 2001.

[16] Enrique Martínez and Gerardo Schneider. Automated analysis of conflicts in software product lines. In *Software Product Lines - 14th International Conference, SPLC 2010, Jeju Island, South Korea, September 13-17, 2010. Workshop Proceedings (Volume 2 : Workshops, Industrial Track, Doctoral Symposium, Demonstrations and Tools)*, pages 75–82, 2010.

[17] Gordon J. Pace, Cristian Prisacariu, and Gerardo Schneider. Model checking contracts - A case study. In *Automated Technology for Verification and Analysis, 5th International Symposium, ATVA*, volume 4762 of *Lecture Notes in Computer Science*.

[18] Gordon J. Pace and Fernando Schapachnik. Permissions in contracts, a logical insight. In *Legal Knowledge and Information Systems - JURIX 2011: The Twenty-Fourth Annual Conference, University of Vienna, Austria, 14th-16th December 2011*, pages 140–144, 2011.

[19] Seyed-Ali Sadat-Akhavi. *Methods of Resolving Conflicts between Treaties*. Number Vol. 3 in Graduate Institute of International Studies (Series). Leiden: Brill Academic Publishers, 2003.

[20] Silvano Colombo Tosatto, Guido Governatori, and Pierre Kelsen. Detecting deontic conflicts in dynamic settings. In *Proceedings of Deontic Logic and Normative Systems DEON*, volume 8554 of *Lecture Notes in Computer Science*.

[21] W. Weber Vasconcelos, M.J. Kollingbaum, and T.J. Norman. Normative conflict resolution in multi-agent systems. *Auton. Agents Multi Agent Syst.*, 19(2):124–152, 2009.

[22] Shufang Zhu, Lucas M. Tabajara, Jianwen Li, Geguang Pu, and Moshe Y. Vardi. A symbolic approach to safety ltl synthesis. In *Hardware and Software: Verification and Testing - 13th International Haifa Verification Conference, HVC 2017*.