Tackling Temporal Deontic Challenges with Equilibrium Logic

Davide Soldà
TU Wien
Vienna, Austria
davide.solda@tuwien.ac.at

Pedro Cabalar
University of A Coruña
A Coruña, Spain
cabalar@udc.es

Agata Ciabattoni
TU Wien
Vienna, Austria
agata@logic.at

Emery Neufeld
TU Wien
Vienna, Austria
emery.a.neufeld@gmail.com

ABSTRACT

Combining temporal reasoning with normative requirements presents significant challenges. In this paper, we tackle the most relevant challenges in the literature from a computational perspective, using Answer Set Programming (ASP). We integrate Temporal Equilibrium Logic, the foundation of Temporal ASP, with Deontic Equilibrium Logic with Explicit Negation, to reason about norms in a temporal context. Our approach is validated by: (i) addressing key benchmarks for temporal normative systems, providing (ii) a normal form reduction that enables the use of existing tools, and (iii) a polynomial LTL reduction for a relevant logic fragment.

KEYWORDS

ASP; Temporal Equilibrium Logic; DELX; Normative Reasoning

ACM Reference Format:
Davide Soldà, Pedro Cabalar, Agata Ciabattoni, and Emery Neufeld. 2025. Tackling Temporal Deontic Challenges with Equilibrium Logic. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Norms — whether social, ethical, or legal — are fundamental to human society, but they are also crucial for the effective functioning of AI agents. Incorporating norms into these agents to guide behavior and foster coordination requires the development of computational frameworks for normative reasoning; such frameworks must account for some non-monotonic features such as defeasibility [44].

Normative reasoning falls under the domain of deontic logic, where obligations and related concepts are at the forefront. In this context, aside from DDL [36], defeasibility has received limited attention, and efficient reasoning tools remain scarce. These features are offered by Answer Set Programming (ASP) — among the most successful paradigms of knowledge representation and reasoning for problem solving [13]. ASP builds on a logic-based rule language, interpreted under answer set semantics for evaluating defeasible negation (negation as failure) [30]. The availability of efficient solvers as well as its solid theoretical foundations have made ASP a popular tool. The logical characterization of ASP based on *Equilibrium Logic* (EL) [45] has also enabled several extensions, including the integration of temporal [3, 21] and deontic operators (e.g. [18, 31–33]). Additionally, normative reasoning has been simulated through encodings in ASP [34, 38] and to the related paradigm of Abductive Logic Programming [41]. Notably, though many practical norms involve deadlines, and normative dynamics often depend on temporal factors, the integration in ASP of temporality *and* norms has been overlooked, with the exception of [32].

Temporal deontic reasoning is inherently complex, even in the context of a single agent. As discussed in [8, 16], beyond the typical intricacies of dealing with norms, such as managing contrary-to-duty (CTD) scenarios — where new obligations arise when other obligations are violated — the temporal dimension introduces new types of norms (e.g., [6, 35]). These include *punctual* obligations, which must be fulfilled at once upon being triggered; *maintenance* obligations, which must be continuously upheld until a specified deadline; and *achievement* obligations, which require meeting a condition at least once before a deadline.

We introduce DeoTEL, a novel non-monotonic framework for temporal deontic reasoning that can be effectively handled by existing ASP solvers. We integrate normative concepts with Temporal Equilibrium Logic (TEL) [2], which combines Linear Temporal Logic (LTL) [43] and EL [45]. Following [18], rather than extending TEL with a modal language, we reason about temporal norms constructed from literals with explicit negation [12]. As a design choice, instead of defining temporal obligations as obligations over temporal formulas, as in [32], we capture them using a single temporal template — the *repeater* — which controls the propagation of obligations over time. Our approach tackles deontic challenges, such as (temporal) CTD norms inducing dilemmas, as in the well-known Gentle Murder paradox [29] (see Example 3.5) where an agent is forbidden to kill, but if it does, it must kill gently, implying both an obligation to kill and a prohibition against it (a dilemma).

We evaluate our framework using established benchmarks arising in defeasible normative reasoning with temporal norms (Section 4), focusing on the single-agent setting to lay a solid foundation for future exploration of more complex multi-agent system (MAS) dynamics. We provide a normal form for the introduced logic, enabling the use of standard ASP tools (e.g., `telingo` [22]) to compute its equilibrium models (Section 5). Although the satisfiability problem for DeoTEL is EXPSPACE-complete, we show (in Section 6) that a PSPACE-complete fragment —capable of representing classical

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

planning domains with deontic preconditions, action effects, and temporal deontic goals— can be polynomially encoded in LTL.

Related Work. Many authors have explored deontic reasoning over time (e.g., [1, 35, 37, 39, 49]; also for MAS [5, 15, 40]), but these lack a clear path to implementation. E.g, temporal defeasible logic has been used for deontic reasoning [35, 37], but a theorem prover is not yet implemented, and this formalism does not include temporal modalities, reasoning instead over specific time points or intervals. DeoTEL inherits the high expressivity of TEL, which can encode the EXSPACE-hard conformant planning problems [11], finding an action sequence that ensures reaching a goal despite incomplete knowledge of the initial state and environmental uncertainties [9, 25]. Such expressivity comes with the price of higher complexity with respect to other temporal deontic formalisms, e.g.[35, 37].

The most similar approach is [32], which integrates temporal and deontic operators with logic programs under answer set semantics, sharing several features with DeoTEL such as the ability to model various types of temporal obligations and non-monotonically cancel them. However, [32] relies on the obligation operator of Standard Deontic Logic (SDL) [51] (the modal logic KD), where the simultaneous obligation and prohibition of the same fact is inconsistent; thus, it cannot handle dilemmas, as in the Gentle Murder paradox. The programs in [32] are based on dynamic LP syntax, while our approach provides a complete semantics for any arbitrary temporal theory, allowing the free combination of temporal, deontic, and LP operators such as rules (seen as implication), default negation, and explicit negation. Moreover, in our formalism, strong equivalence [42] is implied by an equivalence of formulas. The simplicity of DeoTEL limits its ability to derive temporal obligations, which may be seen as a drawback. However, this also safeguards against the unintended derivation of fulfilments and violations from temporal norms that are logically derived but not explicitly stated (Remark 2). Finally, [32] would require an ad hoc implementation, whereas our programs are amenable to existing tools.

2 PRELIMINARIES

Equilibrium Logic has been extended in multiple ways. In this paper, the extension we use as a starting point is Temporal Equilibrium Logic (TEL) [2], in particular, the variant of TEL [3] with the addition of *explicit negation.* Given a countable set of propositional atoms $\mathcal{P}$, a *(temporal) formula* φ is defined by $(p \in \mathcal{P})$

$$\varphi ::= \bot \mid p \mid \sim\varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \wedge \varphi \mid \circ\, \varphi \mid \varphi \;\mathbf{U}\; \varphi \mid \varphi \;\mathbf{W}\; \varphi \quad (1)$$

The connectives $\circ$, $\mathbf{U}$, and $\mathbf{W}$ are temporal operators. The first two, respectively read as "next" and "until", are standard operators in LTL. As usual, $\circ\varphi$ means that φ is true in the next state and $\varphi\mathbf{U}\psi$ means that φ is true until ψ becomes true. The third connective is called "while" and it is known how it can be expressed in terms of $\mathbf{U}$ in LTL but not in TEL. The formula $\varphi\mathbf{W}\psi$ means that φ is true while the condition ψ holds. The formula $\sim\varphi$ is the explicit negation of φ, but we will also handle a (weaker) *default negation* defined as $\neg\varphi := \varphi \rightarrow \bot$. An *(explicit) literal* is an atom $p \in \mathcal{P}$ or its explicit negation $\sim p$. A subset T of Lit, i.e. the set of all explicit literals for atoms in $\mathcal{P}$, is said to be *consistent* if it contains no pair of complementary literals; that is, there is no p for which $\{p, \sim p\} \subseteq T$.

We also consider the following derived operators:

$$\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi) \qquad \top := \sim\bot \qquad \Diamond\varphi := \top \;\mathbf{U}\; \varphi$$
$$\varphi \Leftrightarrow \psi := (\varphi \leftrightarrow \psi) \wedge (\sim\varphi \leftrightarrow \sim\psi)$$
$$\varphi \;\mathbf{R}\; \psi := \sim(\sim\varphi \;\mathbf{U}\; \sim\psi) \qquad \hat{\circ}\varphi := \sim\circ\sim\varphi \qquad \Box\varphi := \bot \;\mathbf{R}\; \varphi$$

Intuitively, $\varphi\mathbf{R}\psi$ is read "release" and means that ψ is true forever or holds until $\varphi \wedge \psi$ becomes true, $\Box\varphi$ means that φ is *always* true, and $\Diamond\varphi$ means that φ is *eventually* true. Meanwhile, the formula $\hat{\circ}\varphi$ is a weaker version of "next" which becomes trivially true if it occurs at the end of a finite *trace.* We sometimes write $\circ^i\varphi$ for $i \geq 0$ defined as $\circ^0\varphi := \varphi$ and $\circ^i\varphi := \circ\circ^{i-1}\varphi$. A *(temporal) theory* is a set of (temporal) formulas; such theories are interpreted over traces.

In standard LTL, a trace $\mathbf{T}$ of length λ is a sequence $\mathbf{T} = (T_i)_{i=0}^{\lambda}$ where each T_i is a propositional interpretation, a subset of atoms $T_i \subseteq \mathcal{P}$ we usually call *state.* We extend this definition in two directions. As a first extension, rather than sets of atoms, states will have the form of consistent sets of explicit literals $T_i \subseteq Lit$. Thus, for instance, we may have a trace of three states like $\mathbf{T} = \{double\} \cdot \{double, \sim smoker\} \cdot \{\sim smoker\}$ where we initially booked a double room with no particular options, then stated we did not want a smoker room, and ended up having no preference about double or single room, while the room should be non-smoking. A second extension is a weakening of certainty related to default reasoning and the model selection criterion we introduce later on. In each state T_i we distinguish a (consistent) subset of literals $H_i \subseteq T_i$ we will say to be "certain", whereas the remaining ones in $T_i \setminus H_i$ hold "by default". Thus, states are now pairs of consistent sets of literals: we may have a state $\langle H_i, T_i\rangle$ with $T_i = \{double, \sim smoker, breakfast\}$ and $H_i = \{double\}$ meaning that *double* $\in H_i$ is certainly true whereas *smoker* and *breakfast* are respectively false and true by default, but none of them are certain. Formally, we define a *trace* as a sequence of pairs $\langle H_i, T_i\rangle$ of consistent sets of explicit literals satisfying $H_i \subseteq T_i \subset Lit$ for every i such that $0 \leq i < \lambda$, where $\lambda \in \mathbb{N} \cup \{\omega\}$ and $\lambda > 0$ is the *length* of the trace. Note that we allow finite $\lambda \in \mathbb{N}$ and infinite $\lambda = \omega$ traces. We write $k \in [a, b)$ for $a \leq k < b$. Alternatively, we also represent the trace as a pair of sequences $\langle \mathbf{H}, \mathbf{T}\rangle$ where $\mathbf{H} = (H_i)_{i\in[0,\lambda)}$ and $\mathbf{T} = (T_i)_{i\in[0,\lambda)}$. A trace is *total* when $\mathbf{H} = \mathbf{T}$, namely, $H_i = T_i$ for all $i \in [0, \lambda)$. For simplicity, a total trace $\langle \mathbf{T}, \mathbf{T}\rangle$ is sometimes written simply as $\mathbf{T}$. Intuitively, in a total trace, all literals are "certain" (none of them holds "by default"). Given a trace $\mathbf{I} = \langle \mathbf{H}, \mathbf{T}\rangle$, we write $\mathbf{I}^t$ to stand for the corresponding total trace $\mathbf{I}^t = \langle \mathbf{T}, \mathbf{T}\rangle$.

The semantics of TEL (with explicit negation) is defined in two steps. First, we define a monotonic logic, *Temporal Here-and-There* (THT) and then, equilibrium models are defined through a model selection criterion among the THT models. The usual semantics for THT is an orthogonal combination of the intermediate logic of *Here-and-There* (HT) (also known as 3-valued Gödel logic) with the standard LTL temporal modalities. The addition of explicit negation further requires enriching the THT satisfaction relation $\models$ with a falsification one $=\!\!|$. The complete description of these two dual relations is shown in the table of Figure 1.

A trace $\mathbf{I}$ is a *model* of a theory Γ, written $\mathbf{I} \models \Gamma$, if $\mathbf{I}, 0 \models \varphi$ for all $\varphi \in \Gamma$. A formula φ is a *tautology* (or *valid*), written $\models \varphi$, iff $\mathbf{I}, i \models \varphi$ for any trace $\mathbf{I}$ and any $i \in [0, \lambda_{\mathbf{I}})$. We call THT the logic induced by the set of all tautologies.

Definition 2.1 (Temporal equilibrium/answer set). A total trace $\langle \mathbf{T}, \mathbf{T}\rangle$ is a *temporal equilibrium model* of a theory Γ if $\langle \mathbf{T}, \mathbf{T}\rangle$ is a

φ	**Satisfaction:** $\mathbf{I}, i \models \varphi$ **when**	**Falsification:** $\mathbf{I}, i \mathrel{=\!\!|} \varphi$ **when**
$\top$ ($\bot$)	always (never)	never (always)
$\alpha \wedge \beta$	$\mathbf{I}, i \models \alpha$ and $\mathbf{I}, i \models \beta$	$\mathbf{I}, i \mathrel{=\!\!\mid} \alpha$ or $\mathbf{I}, i \mathrel{=\!\!\mid} \beta$
$\alpha \vee \beta$	$\mathbf{I}, i \models \alpha$ or $\mathbf{I}, i \models \beta$	$\mathbf{I}, i \mathrel{=\!\!\mid} \alpha$ and $\mathbf{I}, i \mathrel{=\!\!\mid} \beta$
$\alpha \to \beta$	$\mathbf{I}', i \not\models \alpha$ or $\mathbf{I}', i \models \beta$ for $\mathbf{I}' \in \{\mathbf{I}, \mathbf{I}^t\}$	$\mathbf{I}^t, i \models \alpha$ and $\mathbf{I}, i \mathrel{=\!\!\mid} \beta$
$\sim\alpha$	$\mathbf{I}, i \mathrel{=\!\!\mid} \alpha$	$\mathbf{I}, i \models \alpha$
p	$p \in H_i$	$\sim p \in H_i$
$\circ\alpha$	$i + 1 < \lambda_\mathbf{I}$ and $\mathbf{I}, i + 1 \models \alpha$	$i + 1 = \lambda_\mathbf{I}$ or $\mathbf{I}, i + 1 \mathrel{=\!\!\mid} \alpha$
$\alpha \mathbf{U} \psi$	there is some $j \in [i, \lambda_\mathbf{I})$ s.t. $\mathbf{I}, j \models \psi$ and for all $k \in [i, j)$, $\mathbf{I}, k \models \alpha$	for all $j \in [i, \lambda_\mathbf{I})$ either $\mathbf{I}, j \mathrel{=\!\!\mid} \psi$ or there is some $k \in [i, j)$, $\mathbf{I}, k \mathrel{=\!\!\mid} \alpha$
$\alpha \mathbf{W} \psi$	for all $j \in [i, \lambda_\mathbf{I})$ and for $\mathbf{I}' \in \{\mathbf{I}, \mathbf{I}^t\}$ either $\mathbf{I}', j \models \alpha$ or there is $k \in [i, j)$ s.t. $\mathbf{I}', k \not\models \psi$	for some $j \in [i, \lambda_\mathbf{I})$ we have both $\mathbf{I}, j \mathrel{=\!\!\mid} \alpha$ and $\mathbf{I}^t, k \models \psi$ for all $k \in [i, j)$

Figure 1: Semantics of THT (with explicit negation). We use the trace $\mathbf{I} = \langle \mathbf{H}, \mathbf{T} \rangle$, where $\lambda_\mathbf{I}$ is length of the trace.

model of Γ and there is no other model $\langle \mathbf{H}, \mathbf{T} \rangle$ of Γ with $\mathbf{H} \neq \mathbf{T}$. When this happens, we call $\mathbf{T}$ a *temporal answer set* of Γ.

An equilibrium model is a total model $\langle \mathbf{T}, \mathbf{T} \rangle$ where all literals are certain, but there is no "weaker" model of the form $\langle \mathbf{H}, \mathbf{T} \rangle$ with some state $\langle H_i, T_i \rangle$ having a strictly smaller set of certain literals $H_i \subset T_i$. *Temporal Equilibrium Logic (with explicit negation)* (TEL) is the logic induced by temporal equilibrium models.

Example 2.2. Let p be a property varying along time (e.g. the room option we had before), and we add the *inertia* default rules:

$$\Box(p \wedge \neg\circ \sim p \quad \to \quad \circ p) \tag{2}$$

$$\Box(\sim p \wedge \neg \circ p \quad \to \quad \circ \sim p) \tag{3}$$

(2) states that, in any situation where p holds and there is no evidence that its explicit negation must become true in the next state, then p remains true. Formula (3) is analogous for $\sim p$. A theory consisting of (2), (3) and the formula $\circ^2 p \wedge \circ^2 \Diamond \sim p$ has temporal answer sets following the regular expression $\varnothing \cdot \varnothing \cdot \{p\}^+ \cdot \{\sim p\} \cdot \{\sim p\}^\infty$ where ∞ can be $*$ or ω, respectively forming finite or infinite traces.

3 DEONTIC TEMPORAL HERE-AND-THERE

We introduce deontic operators into THT, defining the logic of *Deontic Temporal Here-and-There* (DeoTHT). To this aim, the syntax of temporal formulas (1) is extended with the operators $\mathbf{O}\varphi$ (φ is obligatory) and $\mathbf{F}\varphi$ (φ is forbidden). $\mathcal{L}$ will denote the set of deontic temporal formulas. A *deontic atom* is p, $\mathbf{O}p$, or $\mathbf{F}p$ for any atom $p \in \mathcal{P}$. A (deontic) explicit literal is now any deontic atom A or its explicit negation $\sim A$. By *DLit* we denote the set of all deontic explicit literals (for some signature $\mathcal{P}$); note that $Lit \subseteq DLit$.

The expression $\mathbf{P}\alpha := \sim\mathbf{F}\alpha$ represents the permission for α; $\mathbf{P}\alpha$ can be read as explicit evidence against the prohibition of α. For instance, $\mathbf{P}$*park* means explicit permission to park; that is, stating $\sim\mathbf{F}$*park* we guarantee that there is no prohibition to park. Note the difference with $\neg\mathbf{F}$*park*, which is just the lack of evidence of a prohibition. As before, a set $T \subseteq DLit$ of deontic explicit literals is consistent if there is no deontic atom A for which $\{A, \sim A\} \subseteq T$. Formulas will be interpreted in different contexts or "worlds" depending on their deontic meaning. A *deontic world* [18] w is one of $\{\mathbf{r}, \mathbf{o}, \mathbf{f}\}$ respectively standing for "real", "obligatory" and "forbidden". The complementary world $\bar{w}$ of a world w is: $\bar{\mathbf{o}} = \mathbf{f}$, $\bar{\mathbf{f}} = \mathbf{o}$ and $\bar{\mathbf{r}} = \mathbf{r}$.

DeoTHT-traces will be THT-traces where states become consistent sets of *deontic* literals, i.e., we additionally allow literals of the form $\mathbf{O}p, \mathbf{F}p, \sim\mathbf{O}p$ and $\sim\mathbf{F}p$. However, we also impose the condition below, which affects the $\mathbf{T}$ component in the $\langle \mathbf{H}, \mathbf{T} \rangle$ trace.

Definition 3.1 (Deontic Temporal interpretation). A *deontic temporal interpretation* $\mathbf{T}$ of length λ is a sequence $(T_i)_{i=0}^{\lambda}$ of consistent sets of deontic literals that, for all $p \in \mathcal{P}$ and $i \in [0, \lambda)$, satisfies:

$$\{\mathbf{O}p, \mathbf{F}p\} \subseteq T_i \text{ implies } \{p, \sim p\} \cap T_i \neq \varnothing \tag{4}$$

Condition (4) is a key feature of the deontic extension of Equilibrium Logic with explicit negation in [18] and serves as a relaxation of the standard deontic axiom ($\mathbf{D}$): $\mathbf{O}p \wedge \mathbf{F}p \to \bot$ which disregards the simultaneous obligation and prohibition of the same fact. In our relaxation of ($\mathbf{D}$), an obligation $\mathbf{O}p$ and a prohibition $\mathbf{F}p$ can coexist together at any state T_i, as long as one has been *violated*—either a violated obligation (if $\sim p \in T_i$) or a violated prohibition (if $p \in T_i$). Yet, if neither p nor $\sim p$ is present, we disregard traces where $\mathbf{O}p$ and $\mathbf{F}p$ occur together at the same set T_i.

We can now define deontic traces as we have done in THT, by handling a separation of states formed by certain literals $\mathbf{H}$ among those considered as true $\mathbf{T}$ in the deontic temporal interpretation.

Definition 3.2 (Deontic trace). A *deontic trace* is a triple $\mathbf{I} = \langle \mathbf{H}, \mathbf{T}, w \rangle$ where $w \in \{\mathbf{r}, \mathbf{o}, \mathbf{f}\}$ is a deontic world, $\mathbf{T} = (T_i)_{i \in [0,\lambda)}$ is a deontic temporal interpretation and $\mathbf{H} = (H_i)_{i \in [0,\lambda)}$ is a sequence of sets of deontic literals satisfying $H_i \subseteq T_i \subset DLit$ for all $i \in [0, \lambda)$.

A deontic trace is *total* when $\mathbf{H} = \mathbf{T}$. For any deontic trace $\mathbf{I} = \langle \mathbf{H}, \mathbf{T}, w \rangle$ we denote the associated total trace as $\mathbf{I}^t = \langle \mathbf{T}, \mathbf{T}, w \rangle$.

Definition 3.3. Given a deontic trace $\mathbf{I} = \langle \mathbf{H}, \mathbf{T}, w \rangle$, Figure 2 shows the satisfaction and falsification relations for DeoTHT.

Apart from the addition of the new operators $\mathbf{O}\alpha$ and $\mathbf{F}\alpha$, the only differences from THT (Figure 1) are in the interpretation of an

φ	Satisfaction: $\mathbf{I}, i \models \varphi$ when	Falsification: $\mathbf{I}, i \dashv \varphi$ when
p	$\begin{cases} p \in H_i & \text{if } w = \mathbf{r} \\ \mathbf{O}p \in H_i & \text{if } w = \mathbf{o} \\ {\sim}\mathbf{F}p \in H_i & \text{if } w = \mathbf{f} \end{cases}$	$\begin{cases} {\sim}p \in H_i & \text{if } w = \mathbf{r} \\ {\sim}\mathbf{O}p \in H_i & \text{if } w = \mathbf{o} \\ \mathbf{F}p \in H_i & \text{if } w = \mathbf{f} \end{cases}$
${\sim}\alpha$	$\langle \mathbf{H}, \mathbf{T}, \bar{w} \rangle, i \dashv \alpha$	$\langle \mathbf{H}, \mathbf{T}, \bar{w} \rangle, i \models \alpha$
$\bot, \top, \wedge, \vee, \rightarrow, \circ, \mathbf{U}, \mathbf{W}$	as in Figure 1	as in Figure 1
$\mathbf{O}\alpha$	$\langle \mathbf{H}, \mathbf{T}, o \rangle, i \models \alpha$	$\langle \mathbf{H}, \mathbf{T}, o \rangle, i \dashv \alpha$
$\mathbf{F}\alpha$	$\langle \mathbf{H}, \mathbf{T}, f \rangle, i \dashv \alpha$	$\langle \mathbf{H}, \mathbf{T}, f \rangle, i \models \alpha$

Figure 2: DeoTHT semantics.

atom $p \in \mathcal{P}$ and of explicit negation ${\sim}\alpha$. The interpretation of atoms depends on the deontic world w. For instance, it is easy to see that real-world $\mathbf{r}$ behaves as in THT. However, in the obligation world $\mathbf{o}$, the atom p is satisfied at i if we have a deontic literal $\mathbf{O}p \in H_i$ and falsified if ${\sim}\mathbf{O}p \in H_i$. For the prohibition world $\mathbf{f}$, the situation is dual: p is satisfied at i when ${\sim}\mathbf{F}p \in H_i$ and falsified if $\mathbf{F}p \in H_i$ instead. Notice that the prohibition operator behaves, in this way, as a kind of negation – in fact, we will see that $\mathbf{F}\alpha \equiv \mathbf{O}{\sim}\alpha$. With the interpretation of explicit negation, we still swap the relation $\models$ by $\dashv$ and vice versa, as in THT, but we additionally change the world w to its complementary world $\bar{w}$. Again, when $w = \mathbf{r}$, the semantics collapses to THT, as $\bar{\mathbf{r}}$ is $\mathbf{r}$. For the other two deontic worlds, $\mathbf{o}$ and $\mathbf{f}$, explicit negation alternates from one to another, apart from swapping satisfaction/falsification relations. $\mathbf{O}\alpha$ is satisfied/falsified by "moving" to the $\mathbf{o}$ world whereas, for $\mathbf{F}\alpha$ we both move to the $\mathbf{f}$ world and swap satisfaction with falsification (and vice versa).

A deontic trace $\mathbf{I} = \langle \mathbf{H}, \mathbf{T}, w \rangle$ is a model of a theory Γ, written $\mathbf{I} \models \Gamma$, if $\mathbf{I}, 0 \models \gamma$ for every $\gamma \in \Gamma$. A formula φ is a tautology $\models \varphi$ when any deontic trace $\mathbf{I}$ is a model of φ. Two formulas φ and ψ are said to be *equivalent*, written $\varphi \equiv \psi$, when $\mathbf{I}, i \models \varphi \Leftrightarrow \psi$ for any (finite or infinite) trace $\mathbf{I}$ and any point $i \in [0..\lambda_{\mathbf{I}})$.

The following equivalences (inherited from the non-temporal case [18]) show that the obligation operator $\mathbf{O}$ distributes over the propositional connectives (except explicit negation), and that we can define prohibition $\mathbf{F}$ in terms of obligation $\mathbf{O}$ and vice versa.

$$\begin{array}{cc} \mathbf{O}(\alpha \vee \beta) \equiv \mathbf{O}\alpha \vee \mathbf{O}\beta & \mathbf{O}(\alpha \wedge \beta) \equiv \mathbf{O}\alpha \wedge \mathbf{O}\beta \\ \mathbf{O}(\alpha \rightarrow \beta) \equiv \mathbf{O}\alpha \rightarrow \mathbf{O}\beta & \mathbf{O}\bot \equiv \bot \\ \mathbf{O}\neg\alpha \equiv \neg\mathbf{O}\alpha & \mathbf{O}\mathbf{O}\alpha \equiv \mathbf{O}\alpha \\ \mathbf{O}\,{\sim}\alpha \equiv \mathbf{F}\alpha & \mathbf{F}\,{\sim}\alpha \equiv \mathbf{O}\alpha \end{array}$$

Note how $\mathbf{O}$ differs from a modal necessity operator in that $\mathbf{O}(\alpha \vee \beta)$ can be reduced to $\mathbf{O}\alpha \vee \mathbf{O}\beta$. This behavior may seem a bit strong, but allows us to avoid problems related to entailed obligations like Ross' paradox [46]: if an agent is obliged to send a letter $\mathbf{O}send$ then since $send$ entails $send \vee burn$, the agent is also obliged to send or burn it $\mathbf{O}(send \vee burn)$. Yet, in DeoTHT the latter can be read as $\mathbf{O}send \vee \mathbf{O}burn$ which makes perfect sense. $\mathbf{O}$ also distributes over temporal operators, as stated by the following lemma.

Lemma 3.4 (Perfect recall). *The following hold:*

$$\begin{array}{cc} \mathbf{O} \circ \alpha \equiv \circ\mathbf{O}\alpha & \mathbf{O}(\beta\, \mathbf{U} \alpha) \equiv (\mathbf{O}\beta)\mathbf{U}(\mathbf{O}\alpha) \\ \mathbf{O}(\beta\, \mathbf{R}\, \alpha) \equiv (\mathbf{O}\beta\, \mathbf{R}\, \mathbf{O}\alpha) & \mathbf{O}(\beta\, \mathbf{W}\, \alpha) \equiv (\mathbf{O}\beta\, \mathbf{W}\, \mathbf{O}\alpha) \end{array}$$

$\mathbf{O} \circ \alpha \equiv \circ\mathbf{O}\alpha$ guarantees the *perfect recall* property [14], that is satisfied when we have both the "*no-learning*" property, i.e. $\models \Box(\circ\mathbf{O}\alpha \rightarrow \mathbf{O} \circ \alpha)$, and the "*no-forgetting*" property, i.e., $\models \Box(\mathbf{O} \circ \alpha \rightarrow \circ\mathbf{O}\alpha)$.

Remark 1. *In [14] the authors dropped the* no-learning *property as their orthogonal product of* LTL *with KD could not handle the propagation axiom for achievement obligations. The* no-forgetting *property was instead dropped in [32] to enable cancelation of temporal obligations and their semantics were defined* on top *of* ASP. *As shown in Sec. 4, we can model both cancellation and propagation of temporal norms via a temporal template we call a* repeater.

Apart from permission $\mathbf{P}\alpha := {\sim}\mathbf{F}\alpha$, we introduce the following derived deontic operators (the $*$ superscript means a letter in $\{\mathbf{f}, \mathbf{v}, \mathbf{d}\}$, respectively standing for "fulfilled", "violated" and "defeasible").

$$\begin{array}{ll} \mathbf{O}^{\mathbf{f}}\alpha := \mathbf{O}\alpha \wedge \alpha & \mathbf{O}^{\mathbf{d}}\alpha := \neg\mathbf{P}\,{\sim}\alpha \rightarrow \mathbf{O}\alpha \\ \mathbf{O}^{\mathbf{v}}\alpha := \mathbf{O}\alpha \wedge {\sim}\alpha & \mathbf{F}^{*}\alpha := \mathbf{O}^{*}\,{\sim}\alpha \end{array}$$

Standard Deontic Logic (SDL) [51] is a modal logic that treats the simultaneous obligation $\mathbf{O}\alpha$ and prohibition $\mathbf{F}\alpha$ of the same formula α as inconsistent, formalized by the (**D**) axiom we introduced before and that can also be stated as $\mathbf{O}\alpha \rightarrow {\sim}\mathbf{O}{\sim}\alpha$ using the relation $\mathbf{F}\alpha \equiv \mathbf{O}\,{\sim}\alpha$. The deontic extension of equilibrium logic [18] introduced a weaker variant of this axiom (**wD**) ensuring inconsistency arises only when neither an obligation nor a prohibition is violated.

This idea is behind Condition (4) of Def. 3.1 and can be captured by the addition of the following temporal (**wD**) axiom schema:

$$\Box((\mathbf{O}p \wedge \mathbf{F}p \wedge \neg p \wedge \neg\, {\sim}p) \rightarrow \bot) \qquad (\mathbf{TwD})$$

for every (non-deontic) atom $p \in \mathcal{P}$. (**TwD**) disallows *dilemmas*, such as $\mathbf{O}k \wedge \mathbf{F}k$ — simultaneous obligations to kill k and not kill ${\sim}k$, which remain unresolved without further information. However, to accommodate certain CTD scenarios, (**TwD**) relaxes deontic inconsistency, as shown below.

Example 3.5. Recall the Gentle Murder paradox [29], where killing is prohibited $\Box\mathbf{F}k$, but if done, must be gentle, $\Box(k \rightarrow \mathbf{O}g)$, where $\Box(\mathbf{O}g \rightarrow \mathbf{O}k)$. From the fact that an agent kills in, say, the second state $\circ k$, we have $\circ(\mathbf{O}k \wedge \mathbf{F}k)$, a deontic inconsistency which is disallowed by formalisms like [32]. However, by adopting (**TwD**) instead of (**D**), $\circ k$ causes us not to derive $\bot$ (since $\circ\mathbf{F}k$ has been violated), accommodating the CTD scenario.

Theorem 3.6 (Conservative extension). *Given a theory* Γ *containing no temporal operators and with only* $\mathbf{O}$ *as a deontic operator,* $\mathbf{T} \models \Gamma$ *in* DeoTHT *if and only if* $T_0 \models \Gamma$ *in Deontic* HT *(DHTX) according to Defn. 1 in [18].*

Persistence, a key property in intuitionistic logic, ensures that accessible worlds (here, the $\mathbf{T}$ component) validate or refute at least the same formulas as the current world (the $\mathbf{H}$ component).

Theorem 3.7. *For each trace* $\mathbf{I} = \langle \mathbf{H}, \mathbf{T}, w \rangle$, $i \in [0..\lambda_{\mathbf{I}})$ *and* $\varphi \in \mathcal{L}$: $\mathbf{I}, i \models \varphi$ *implies* $\mathbf{I}^t, i \models \varphi$ *and* $\mathbf{I}, i \mathrel{=\!|} \varphi$ *implies* $\mathbf{I}^t, i \mathrel{=\!|} \varphi$.

Finally, we extrapolate Defn. 2.1 for the deontic extension.

*Definition 3.8 (*DeoTEL*).* A total deontic trace $\langle \mathbf{T}, \mathbf{T}, r \rangle$ is a *deontic temporal equilibrium model* of a theory Γ if $\langle \mathbf{T}, \mathbf{T}, r \rangle$ is a model of Γ and there is no other model $\langle \mathbf{H}, \mathbf{T}, r \rangle$ of Γ with $\mathbf{H} \neq \mathbf{T}$. When this happens, we call $\mathbf{T}$ a *deontic temporal answer set* of Γ. DeoTEL is the logic induced by *deontic temporal equilibrium models.*

Note that the evaluation of formulas in a theory starts in the real world $\mathbf{r}$ (and at state 0).

Example 3.9. Let $\Gamma = \{\Box \mathbf{F}k, \Box(k \to \mathbf{O}g), \Box(\mathbf{O}g \to \mathbf{O}k), \circ k\}$ from Ex. 3.5. A deontic temporal equilibrium model for Γ is $\langle \mathbf{T}, \mathbf{T}, r \rangle$ where $\mathbf{T} = \{\mathbf{F}k\} \cdot \{k, \mathbf{F}k, \mathbf{O}k, \mathbf{O}g\} \cdot \{\mathbf{F}k\}^*$. Note that while $\mathbf{O}k, \mathbf{F}k \in T_1$, as $k \in T_1$, condition 4 of Def. 3.1 is met and it is a deontic trace.

4 TEMPORAL NORMS IN DeoTEL

We demonstrate the capabilities of our logic by addressing some key challenges of deontic reasoning in a temporal context. We examine various types of norms; consider the temporal adaptation below of the Chisholm paradox [23, 24], with which we model pandemic restrictions for a human agent:

(A) If you test positive, you ought to remain in quarantine until you test negative.
(B) If you test positive and are in quarantine, you ought to report your quarantine before you test negative.
(C) While you are violating an obligation to quarantine, it is forbidden to report you are in quarantine.

This is a CTD scenario; it contains both a primary maintenance obligation (A) and a secondary punctual obligation (C) of not reporting while the agent violates (A). Meanwhile, the achievement obligation (B) potentially leads to contradictory indications (i.e. a *dilemma*), if you are not in quarantine.

Formalizing **punctual obligations** (e.g. (C)) is straightforward, as it collapses to the non-temporal case; if an obligation to *q*uarantine is violated $\mathbf{O}^{\mathbf{v}}q$, it is forbidden to *r*eport a quarantine $\mathbf{F}\ r$. This conditional norm can be represented in the temporal context as

$$\Box(\mathbf{O}^{\mathbf{v}}q \to \mathbf{F}\ r) \qquad (C)$$

a global implication equivalent to $\Box(\mathbf{O}q \wedge {\sim}q \to \mathbf{O}\ {\sim}r)$.

Maintenance and achievement obligations require more attention. Recall that, once triggered, **maintenance obligations** persist until the deadline arrives so they can be violated (or fulfilled) at each point up to the deadline, suggesting that we can model them by propagating a series of punctual obligations up to a given deadline. Meanwhile, **achievement obligations**, which are only violated when not fulfilled by the deadline and can be fulfilled only when the obligatory condition is achieved (by the deadline), prove a greater challenge. If we are employing punctual obligations in the modeling of achievement obligations, the punctual obligation should only appear once – when the obligation is fulfilled or violated.

One might consider modeling such obligations using $\mathbf{U}$, $\mathbf{R}$, or $\mathbf{W}$ over punctual obligations. However, $\mathbf{U}$ and $\mathbf{R}$ are unsuitable to represent temporal obligations whose deadlines may never arrive, and $\mathbf{W}$ might lead to undesired behaviors.

Remark 2. *Considering maintenance and achievement obligations as obligations of temporal formulas* $\mathbf{O}\alpha$ *(as in [32]) causes issues regarding entailed obligations as in Ross' paradox [46]. For instance, suppose* $\alpha = pay\ \mathbf{R} \sim january$*, so* $\mathbf{O}\alpha$ *obliges paying by January. This entails* $\mathbf{O}(pay\mathbf{R} \sim\delta)$ *for any later deadline* δ *or in general, any formula entailed by* α*. In that way, if we fulfill* $\mathbf{O}\alpha$ *we also fulfill many other entailed obligations that were* never stated explicitly *(the same applies for violations), making it impossible to, e.g., count how many obligations have been fulfilled/violated.*

With these complications in mind, we employ a technique similar to predicate abstraction but for operators. We encode them via atoms that work as parameterized templates governed by the violation/fulfillment/propagation conditions of achievement and maintenance obligations. To that end, we enrich our set of propositional variables $\mathcal{P}$ with a set of designed atoms defined below:

Definition 4.1 (Repeater). Given any DeoTEL formula α over atoms $\mathcal{P}$ and $S \subseteq Lit$, we write $\mathbf{Rep}_S(\alpha)$ to stand for a new atom (not included in $\mathcal{P}$) called *repeater.*

When extending the signature with repeaters, we also restrict the syntax so that repeaters do not appear within the scope of $\mathbf{O}$. Repeaters have a semantics governed by the following axioms:

$$\Box(\mathbf{Rep}_S(\alpha) \to \alpha) \qquad \text{(Deriv.)}$$

$$\Box(\neg\hat{\circ} \sim\mathbf{Rep}_S(\alpha) \wedge \mathbf{Rep}_S(\alpha) \wedge \neg \bigvee S \to \hat{\circ}\mathbf{Rep}_S(\alpha)) \qquad \text{(Prop.)}$$

The (Deriv.) rule is responsible for the derivation of α when the repeater $\mathbf{Rep}_S(\alpha)$ is in force, while rule (Prop.) is responsible for the propagation of the repeater until one of the stopping conditions in S is met, or when ${\sim}\mathbf{Rep}_S(\alpha)$ is derived at the next state. We call ${\sim}\mathbf{Rep}_S(\alpha)$ the *cancellation* of $\mathbf{Rep}_S(\alpha)$. Note that the ability to *cancel* a temporal obligation is another potential advantage of this approach over defining temporal obligations via a temporal operator such as $\mathbf{U}$. Using the repeaters, we propose a novel definition of achievement and maintenance obligations. For every $l, \delta \in Lit$:

$$\mathbf{A}_\delta(l) := \mathbf{Rep}_{\{\delta,l\}}((\delta \vee l) \to \mathbf{O}l) \qquad \mathbf{M}_\delta(l) := \mathbf{Rep}_{\{\delta\}}(\mathbf{O}l) \qquad (5)$$

We can also define their defeasible versions (obligations that can be overridden) $\mathbf{A}^{\mathbf{d}}_\delta(l)$ and $\mathbf{M}^{\mathbf{d}}_\delta(l)$ by replacing $\mathbf{O}l$ with $\mathbf{O}^{\mathbf{d}}l$ in the respective definitions above.

Remark 3. *Using repeaters enables propagating maintenance obligations even after they are violated the first time, unlike [31].*

To illustrate how we model temporal obligations with repeaters, let us return to (A-B). (A) is a maintenance obligation, which obligates you to, upon testing positive p, maintain a state of quarantine q until you test negative n. This conditional obligation is written as:

$$\Box(p \to \mathbf{M}_n(q)) \qquad (A)$$

That is, whenever p occurs, $\mathbf{M}_n(q)$ is triggered, and is propagated until n occurs, and $\mathbf{O}q$ is derived every step along the way due to (Deriv.). Meanwhile, achievement (B) can be modeled as:

$$\Box((p \land q) \to \mathbf{A}_n(r)) \qquad (B)$$

This means that the trigger $p \land q$ prompts the propagation of $\mathbf{A}_n(r)$ until $n \lor r$ occurs — until then, we derive $(n \lor r) \to \mathbf{O}r$. Thus, if r does not occur before n, then with n we derive $\mathbf{O}r$, and depending on whether or not r is true, we furthermore have $\mathbf{O^f}r$ or $\mathbf{O^v}r$.

With the repeater mechanism, we can also define temporal permissions. [35] characterizes **maintenance permissions** as permission to remain in a certain state for a certain period. Such a permission can be exercised any number of times before a deadline occurs; these permissions can be modeled by propagating punctual permissions until a deadline; in other words, $\mathbf{P}^{\mathbf{m}}_{\delta}(l) := \mathbf{Rep}_{\{\delta\}}(\mathbf{P}l)$. **Achievement permissions**, on the other hand, can only be exercised once before a deadline. [35] exemplifies achievement permissions as: the right to return a product before a deadline occurs. It is permissible (as in, there is a punctual permission) to return the product at each time step, up until the point of time when you return the product, or until the deadline (if you don't return it). Achievement permissions can be simply modeled as $\mathbf{P}^{\mathbf{a}}_{\delta}(l) := \mathbf{Rep}_{\{\delta,l\}}(\mathbf{P}l)$.

4.1 Selected Addressed Challenges

We discuss how DeoTEL handles established benchmarks for defeasible normative reasoning in time; we will present temporal variants of the challenges from [18] (those where adding time causes notable changes), which were in turn based on the discussion in [16].

Temporal CTDs and Dilemmas. Consider the normative system (A-C). The maintenance obligation (A) is triggered upon testing positive, the achievement obligation (B) is triggered by both testing positive and being (initially) in quarantine, and (C) is a CTD obligation w.r.t. (A). The addition of (C) to (A-B) introduces the potential for a dilemma to occur; if you test positive p at some point in the trace and initially are in quarantine, obligations (A-B) are triggered. (B) is an achievement obligation and can be neglected temporarily, but when the deadline n arrives, that is your last chance to fulfill it, and thus you are required to report your quarantine ($\mathbf{O}r$ is derived with (Deriv.)). However, if you are *not actually in quarantine*, obligation (C) is in force, requiring that you *don't* report your quarantine ($\mathbf{F}r$). Thus, if you put off reporting your quarantine until the deadline, and are not in quarantine at that point, it is both forbidden and obligatory to report your quarantine.

To solve the deontic inconsistency that arises from this dilemma, we need explicit information on whether or not you report your quarantine at deadline n, ensuring that $r \lor \sim r$ holds. Then, the temporal weak D axiom (**TwD**) is not triggered, and $\bot$ is not derived.

Compensatory Obligations. We can add to (A-C) another obligation that has a specific role: compensating for the violation of another obligation. In this case, we add the achievement obligation stating that an obligation to pay a fine f by the end of the month m arises from violating the obligation to quarantine, or:

$$\Box(\mathbf{O^v}q \to \mathbf{A}_m(f)) \qquad (D)$$

Thus, if (A) is in force and then violated, since $\Box(\mathbf{M}_n(q) \to \mathbf{O}q)$, the achievement obligation $\mathbf{A}_m(f)$ is triggered as well, and will also be violated if the fine is not paid by the end of the month.

Cancellation of Temporal Obligations. A maintenance or achievement obligation that is in force may, for various reasons, end up voided or canceled. To demonstrate this, let us add another norm to (A-C): if the lockdown ends l_e, the obligations to remain in quarantine and report your quarantine are canceled. That is

$$\Box(l_e \to \sim\mathbf{M}_n(q)),\ \Box(l_e \to \sim\mathbf{A}_n(r)) \qquad (E)$$

With this cancellation norm in place, if we have a trace where p is true at some point, triggering $\mathbf{M}_n(q)$ and $\mathbf{A}_n(r)$, and l_e occurs before n, the rule (Prop.) cannot be used to propagate $\mathbf{M}_n(q)$ or $\mathbf{A}_n(r)$ to the next state. Note that canceling obligations is only possible when we use the repeater to model temporal obligations.

Defeasible Obligations. Some obligations are defeasible and act as defaults, which may be overruled by another norm, granting an exception. For example, let us replace (A) above with (A'): unless you are explicitly permitted to do otherwise, upon testing positive p you must quarantine q until you test negative n. Then we might have a permission that acts as an exception to this obligation: it is permissible to leave quarantine $\sim q$ in the case of an emergency e, until the situation is solved s (a maintenance permission); i.e.:

$$\Box(p \to \mathbf{M}^{\mathbf{d}}_n(q)) \qquad (A') \qquad\qquad \Box(e \to \mathbf{P}^{\mathbf{m}}_s(\sim q)) \qquad (F)$$

Thus, even when (A') is in force, if an emergency e occurs, $\mathbf{P}^{\mathbf{m}}_s(\sim q)$ temporarily comes into force until s, overriding (A').

Constitutive Norms. Constitutive norms [47] of the form "X counts as Y in context C" can be modeled in our temporal framework straightforwardly. To put an example, the sentence "a quarantine (q) counts as a confinement (cn)" can be represented as:

$$\Box(q \to cn),\ \Box(\mathbf{O}q \to \mathbf{O}cn),\ \Box(\sim cn \to \sim q),\ \Box(\mathbf{O}\sim cn \to \mathbf{O}\sim q)$$

so that being forced to be in quarantine is also being forced to confine, and similarly, having a prohibition to confine implies a prohibition to quarantine. These constitutive norms can also be transformed into defaults applicable only under non-violation, such as $\Box(\mathbf{O}q \land \neg \sim q \to \mathbf{O}cn)$ (see [18] for further details).

5 DEONTIC TEMPORAL LOGIC PROGRAMS

We introduce a procedure that enables us to reduce a generic DeoTHT formula into a set of Deontic Temporal rules. A *temporal literal* L has one of the forms $d, \circ d, \hat{\circ} d$ for any $d \in DLit$.

Definition 5.1 (DeoTLP). A Deontic Temporal program comprises

- *initial* rules of the form (all b_i, c_j are temporal literals)
$$r : (\neg)b_1 \land \cdots \land (\neg)b_k \to (\neg)c_1 \lor \cdots \lor (\neg)c_l; \qquad (6)$$
- *dynamic* rules of the form $\Box r$, where r is an initial rule;
- *fulfillment* rules of either the form $\Box(\Box p \to q)$ or $\Box(p \to \Diamond q)$, where $p, q \in DLit$.

An initial or dynamic rule r is a *constraint* if its head is $\bot$; it is a *fact* if its body is empty (k=0) and its head is a single positive literal.

Our normal form extends the one for TEL [17], by incorporating explicit negation and deontic modalities according to our semantics (Defn. 3.3). As in [17], our reduction follows two steps. First, we make a Tseitin-like [50] transformation, $df'(\gamma)$ that replaces each subformula γ by an auxiliary atom $\mathbf{L}_\gamma$. We call such an alphabet $\mathcal{P}_\mathbf{L}$. This produces a theory that is not in *DeoTLP* form yet, but the latter is achieved through a second step of transformations $df^*(df'(\gamma))$. The whole transformation is polynomial and the result is strongly equivalent to the original theory (modulo auxiliary atoms).

In what follows, we assume $\mathbf{L}_\phi := \phi$ for formulas $\phi \in DLit \cup \{\top, \bot\}$, so we do not introduce extra atoms in those cases.

Definition 5.2 (Transformation $df(\cdot)$). Given a formula γ, $df(\gamma) := df^*(df'(\gamma))$ where $df'(\cdot)$ and $df^*(\cdot)$ are defined in Table 1.

The transformation $df^*(\cdot)$ is as in [17], except for the case that unfolds an implication in the consequent of another implication (leftmost, bottom in Table 1). The addition of the formula ψ in the disjunction was not needed in [17] but it is, in the presence of explicit negation, to maintain an equivalent result. We can state:

Theorem 5.3. *Given a theory* Γ, $\{\mathbf{I} \mid \mathbf{I} \models \Gamma\} = \{\mathbf{I}' \cap DLit \mid \mathbf{I}' \models \Gamma'\}$ *where* Γ' *is the DeoTLP program* $\Gamma' = \{\mathbf{L}_\gamma \mid \gamma \in \Gamma\} \cup \{df(\phi) \mid \phi$ *subformula of* $\Gamma\}$.

Proof (sketch). We (i) show the correctness of the df' transformation, and (ii) verify the correctness of the df^* rewriting. (i) We demonstrate that a DeoTHT model $\mathbf{I}$ exists for Γ iff there is a DeoTHT model for $\Gamma' = df'(\Gamma)$, where the extended alphabet is $\mathcal{P}_\mathbf{L}$. We prove that the newly introduced atoms are true if and only if so are the formulas they represent in a Tseitin-like manner. This can be verified through a structural induction on the formulas. (ii) The correctness of the df^* rewriting is established using truth tables, proving that Γ' and $df^*(\Gamma')$ have the same DeoTHT models. □

Since our target semantics is DeoTEL, we stress the following Corollary of Theorem 5.3.

Corollary 5.4. *Given a theory* Γ, $\{\mathbf{T} \mid \mathbf{T}$ *is a* DeoTEL *model of* $\Gamma\} = \{\mathbf{T}' \cap DLit \mid \mathbf{T}'$ *is a* DeoTEL *model of* $\Gamma'\}$ *where* Γ' *is the DeoTLP program* $\Gamma' = \{\mathbf{L}_\gamma \mid \gamma \in \Gamma\} \cup \{df(\phi) \mid \phi$ *sub-formula of* $\Gamma\}$.

6 COMPUTATION AND COMPLEXITY

We study the complexity and computation of DeoTEL, providing two polynomial reductions: one reduces DeoTEL, whose satisfiability problem is EXPSPACE-complete, into TEL; another targets a practical fragment of the formalism, which can be encoded into LTL. Furthermore, the translation into TEL reduces a theory into a temporal logic program. This permits the use of efficient solvers, such as `telingo` [22].

Proposition 6.1. *Given a set of atoms* $\mathcal{P}$, *let* $\mathcal{P}'$ *denote the extended set* $\mathcal{P}' := \mathcal{P} \cup \{\mathbf{O}p, \mathbf{F}p \mid p \in \mathcal{P}\}$ *and let* $\mathbf{T} = (T_i)_{i=0}^{\lambda}$ *conform a* THT*-total trace* $\langle \mathbf{T}, \mathbf{T} \rangle$ *for signature* $\mathcal{P}'$. *Then, for any temporal theory* Γ *where* $\mathbf{O}$ *and* $\mathbf{F}$ *are only applied to atoms:*

$$\mathbf{T} \models \Gamma \text{ in DeoTHT iff } \mathbf{T} \models \Gamma \cup \{(\mathbf{TwD})\} \text{ in THT.}$$

We need an intermediate result to establish the membership of satisfiability for a DeoTEL theory under DeoTEL semantics.

Theorem 6.2. *Given a DeoTLP program* π, *there is a polynomial reduction to a theory* π' *such that* π *has a* DeoTEL *model if and only if* π' *has a* TEL *model.*

Proof (sketch). Define $\mathcal{P}_D$ as $DLit \setminus \{{\sim}L \mid {\sim}L \in DLit\}$. We transform the *DeoTLP* program π in four steps. We rewrite (i) every $\mathbf{O}(p)$ (resp. $\mathbf{F}(p)$) into $p_\mathbf{o}$ (resp. $p_\mathbf{f}$), and (ii) every literal ${\sim}p$ (including the fresh ones) with a fresh atom $\bar{p}$; we add (iii) the constraint $\Box(p \wedge \bar{p} \rightarrow \bot)$ for every atom p (including the fresh ones); and (iv) the constraint $\Box((p_\mathbf{o} \wedge p_\mathbf{f} \wedge \neg p \wedge \neg\bar{p}) \rightarrow \bot)$ for each $p \in \mathcal{P}_D$. Step (i) reduces modalities to atoms. It is easy to show that $\mathbf{I}$ is a DeoTHT model for the original program π iff $\mathbf{I}'$ where $\mathbf{O}(p)$ are replaced by $p_\mathbf{o}$ is a DeoTHT model of the rewritten program after step (i). Steps (ii) and (iii) use a standard technique that rewrites the explicit negation of an atom ${\sim}p$ into a fresh atom $\bar{p}$, and ensures the preservation of the same models. Finally, thanks to Prop. 6.1 step (iv) rules out interpretations that are not temporal deontic. □

On the computational side, as a corollary of this theorem, follows a procedure to encode DeoTEL theories into TEL temporal programs, enabling the use of existing tools for computing models.

Theorem 6.3. *Satisfiability of* DeoTEL *is EXSPACE-complete.*

Proof. Given a DeoTEL theory Γ, we reduce it into a *DeoTLP* program π as done in Th. 5.3. π can be rewritten into π' as stated in Th. 6.2. Deciding weather a theory Γ has a TEL model is in EXSPACE [19]. The hardness for the infinite case follows by Th. 3.6 and Th. III.1 in [10], which can be adapted to prove the hardness for the finite trace semantics, as noted in [20]. □

To tame the computational complexity of our formalism, we identify a PSPACE-complete fragment of DeoTEL that can be polynomially encoded into LTL. The fragment allows us to encode classical planning problems. We introduce below the deontic version of the Splittable Temporal Logic Program (*SDeoTLP*) fragment of temporal programs as defined in [4]. This fragment offers computational advantages by enabling the splitting of the unfolded program into slices, which allows the DeoTEL models to be built in a step-wise fashion by concatenating slices of local answer sets.

Definition 6.4 (Splittable DeoTLP (SDeoTLP)). *SDeoTLP* consists of rules of types (i_0), (i_1), and (d), defined as:

- (i_0) $(\neg)b_1 \wedge \dots (\neg)b_k \rightarrow c_1 \vee \dots \vee c_l$ with $k, l \geq 0$ and $b_i, c_j \in DLit$,
- (i_1) $(\neg)b_1 \wedge \dots (\neg)b_k \rightarrow c_1 \vee \dots \vee c_l$ with $k, l \geq 0$, $b_i \in DLit \cup \circ DLit \cup \hat{\circ} DLit$ and $c_i \in \circ DLit \cup \hat{\circ} DLit$,
- (d) $\Box(r)$ where r is a rule of type (i_1).

Rules (i_0) and (i_1), called initial rules and denoted by $ini(\pi)$, correspond to initial rules for situation 0 and to the transition to situation 1, respectively; Rules of type (d), called dynamic rules and denoted by $dyn(\pi)$, are meant to hold at every position along the trace.

We define temporal literals as $\{p, \circ\, p, \hat{\circ}\, p \mid p \in DLit\}$, where $B^+(r)$ and $B^-(r)$ denote the set of temporal literals and default negated literals in the body of the rule r, and $H(r)$ denotes the set $\{c_1, \dots, c_l\}$. To prove the LTL reduction, we formalize dependencies among literals using a dependency graph. Given a *SDeoTLP* program π, its 2-unfolded version π^2 contains every initial rule in $init(\pi)$ and for each dynamic rule $r \in dyn(\pi)$, it contains $\circ^i(r)$ for all $0 \leq i < 3$. Given an unfolded rule $\circ^k(r)$, we denote $p, \circ\, q \in B^+(r) \cup B^-(r) \cup H(r)$ as p^k and q^{k+1}.

Definition 6.5 (Dependency Graph). The *dependency graph* of a *SDeoTLP* program π is the graph $DG_\pi = \langle V, E \rangle$ where $V = \{p^i \mid p \in$

Table 1: Definition of $df'(\cdot)$ and $df^*(\cdot)$. $[\sim]$ means that the explicit negation is optional.

$$
\begin{array}{ll}
df'(\gamma) = df'(\phi) & \text{if } \gamma = \sim\sim \phi \\
df'(\gamma) = \Box\,(\mathbf{L}_\gamma \leftrightarrow \oplus\mathbf{L}_\phi) \wedge \Box\,(\sim\mathbf{L}_\gamma \leftrightarrow \sim\oplus\mathbf{L}_\phi) & \text{if } \gamma = [\sim] \oplus \phi \text{ with } \oplus \in \{\mathbf{O}, \mathbf{F}\} \\
df'(\gamma) = \Box\,(\mathbf{L}_\gamma \leftrightarrow \mathbf{L}_\phi \oplus \mathbf{L}_\psi) \wedge \Box\,(\sim\mathbf{L}_\gamma \leftrightarrow \sim\mathbf{L}_\phi \otimes \sim\mathbf{L}_\psi) & \text{if } \gamma = [\sim](\phi \oplus \psi) \text{ with } (\oplus, \otimes) \in \{(\vee, \wedge), (\wedge, \vee)\} \\
df'(\gamma) = \Box\,(\mathbf{L}_\gamma \leftrightarrow \mathbf{L}_\phi \rightarrow \mathbf{L}_\psi) \wedge \Box\,(\sim\mathbf{L}_\gamma \leftrightarrow \neg\neg\mathbf{L}_\phi \wedge \sim\mathbf{L}_\psi) & \text{if } \gamma = [\sim](\phi \rightarrow \psi); \\
df'(\gamma) = \Box\,(\mathbf{L}_\gamma \leftrightarrow \oplus\mathbf{L}_\phi) \wedge \Box\,(\sim\mathbf{L}_\gamma \leftrightarrow \otimes \sim\mathbf{L}_\phi) & \text{if } \gamma = [\sim](\oplus\psi) \text{ with } (\oplus, \otimes) \in \{(\circ, \hat{\circ}), (\hat{\circ}, \circ)\} \\
df'(\gamma) = \Box\,(\mathbf{L}_\gamma \leftrightarrow (\mathbf{L}_\phi \wedge (\mathbf{L}_\psi \rightarrow \hat{\circ}\mathbf{L}_\gamma))) & \\
\wedge\, \Box\,(\sim\mathbf{L}_\gamma \leftrightarrow \sim\mathbf{L}_\phi \vee (\neg\neg\mathbf{L}_\phi \wedge \circ \sim\mathbf{L}_\gamma)) \wedge \Box\,(\sim\mathbf{L}_\gamma \rightarrow \Diamond \sim\mathbf{L}_\phi) & \text{if } \gamma = [\sim](\phi\ \mathbf{W}\ \psi) \\
df'(\gamma) = \Box\,(\mathbf{L}_\gamma \leftrightarrow \mathbf{L}_\psi \vee (\mathbf{L}_\phi \wedge \circ\mathbf{L}_\gamma)) \wedge \Box\,(\mathbf{L}_\gamma \rightarrow \Diamond\, \mathbf{L}_\psi) & \\
\wedge\, \Box\,(\sim\mathbf{L}_\gamma \leftrightarrow \sim\mathbf{L}_\psi \wedge (\sim\mathbf{L}_\phi \vee \hat{\circ} \sim\mathbf{L}_\gamma)) \wedge \Box\,(\Box \sim\mathbf{L}_\psi \rightarrow \sim\mathbf{L}_\gamma) & \text{if } \gamma = [\sim](\phi\ \mathbf{U}\ \psi)
\end{array}
$$

$$df^*((\phi \vee \psi) \wedge \alpha \rightarrow \beta) = (\phi \wedge \alpha \rightarrow \beta) \wedge (\psi \wedge \alpha \rightarrow \beta)$$

$$df^*(\neg\neg\phi \wedge \alpha \rightarrow \beta) = \alpha \rightarrow \neg\phi \vee \beta$$

$$df^*(\alpha \rightarrow (\phi \rightarrow \psi) \vee \beta) = \begin{cases} (\phi \wedge \alpha \rightarrow \psi \vee \beta) \quad \wedge \\ (\neg\psi \wedge \alpha \rightarrow \psi \ \vee \neg\phi \vee \beta) \end{cases}$$

$$df^*(\alpha \rightarrow (\phi \wedge \psi) \vee \beta) = (\alpha \rightarrow \phi \vee \beta) \wedge (\alpha \rightarrow \psi \vee \beta)$$

$$df^*(\alpha \rightarrow \neg\neg\phi \vee \beta) = \neg\phi \wedge \alpha \rightarrow \beta$$

$$df^*((\phi \rightarrow \psi) \wedge \alpha \rightarrow \beta) = \begin{cases} (\alpha \rightarrow \phi \vee \neg\psi \vee \beta) \wedge \\ (\neg\phi \wedge \alpha \rightarrow \beta) \wedge (\psi \wedge \alpha \rightarrow \beta) \end{cases}$$

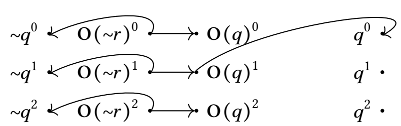

Figure 3: Temporal dependency graph of Π from Example 6.7.

DLit and $i = 0, 1, 2\}$ and $(a, b)^+ \in E$ (resp. $(a, b)^- \in E$) if $a \in H(r)$ and $b \in B^+(r)$ (resp. $b \in B^-(r)$) for some $\circ^k r$ in π^2.

The 2-unfolding suffices because dependencies in the graph link each position only to itself or the previous one, remaining unchanged from position 2 onward. For a general *DeoTLP* program this assumption does not hold as shown in the example below.

Example 6.6. [Non *SDeoTLP* program] Let π be $\{\Box(p \rightarrow \circ q), \Box(q \rightarrow \circ r), \Box(r \rightarrow \circ s), \Box(\circ s \rightarrow s), \Box(\circ p \rightarrow s)\}$, the edges of DG^3_π are $(p^0, s^1), (s^1, s^2), (s^2, s^3), (r^2, s^3), (q^1, r^2), (q^1, p^0))$ which form loops that the 2-unfolded dependency graph would not detect.

Example 6.7 (SDeoTLP program). Let us consider the *SDeoTLP* program consisting of the (d)-rule (C) from Section 4, the (i_0)-rule q, and the (i_1)-rule $q \rightarrow \circ\mathbf{O}(q)$. The initial rule (i_0) has an empty body and counts as fact. Let us denote this theory as Π, with its dependency graph DG_Π illustrated in Figure 3, which is cropped to display only the relevant deontic literals. Due to the syntactic constraints of *SDeoTLP* programs, all loops can be identified within the 2-unfolded dependency graph.

The following theorem is an extension of a result in [2], whose proof leverages the fact that the model construction can be performed slice-wise via the generalized Lin–Zhao theorem [28].

Theorem 6.8. *Any SDeoTLP program π, can be encoded in* LTL *in such a way that π is* LTL*-satisfiable iff π admits a* DeoTEL *model.*

To obtain a polynomial LTL encoding we introduce another syntactic constraint (still capturing all the examples in this paper). Given a disjunction-free *SDeoTLP* program, and its dependency graph $DG_\pi = \langle V, E \rangle$, if there exists a total order $v : V \rightarrow V$ s.t. if $l < l'$ then there is no positive loop in DG_π from l to l', we say that π is *tight*.

Theorem 6.9. *Given a tight SDeoTLP program π and a* DeoTEL *formula ϕ, deciding whether $\pi \wedge \neg\phi$ is satisfiable is PSPACE-complete and its* LTL *encoding is polynomial.*

This result is achieved by leveraging the slice-based encoding of splittable programs into LTL, and the fact that the tightness of the slices guarantees a polynomial encoding via a completion technique [27]. As an example of completion for tight *DeoTLP*, consider the program Π from Example 6.7. Π can be encoded into LTL via completion as $s_0 \wedge s_1 \wedge \circ\Box(s_2)$, where:

$$s_0 = q \ \wedge\ (r\bar{q}_{\mathbf{o}} \leftrightarrow (q_{\mathbf{o}} \wedge \bar{q})) \wedge \neg\bar{q} \wedge \neg q_{\mathbf{o}}$$

$$s_1 = \neg \circ q \ \wedge\ (\circ r\bar{q}_{\mathbf{o}} \leftrightarrow (\circ q_{\mathbf{o}} \wedge \circ\bar{q})) \wedge \neg \circ \bar{q} \wedge (\circ q_{\mathbf{o}} \leftrightarrow q)$$

$$s_2 = \neg \circ q \ \wedge\ (\circ r\bar{q}_{\mathbf{o}} \leftrightarrow (\circ q_{\mathbf{o}} \wedge \circ\bar{q})) \wedge \neg \circ \bar{q} \wedge \neg \circ q_{\mathbf{o}}$$

The significance of tight logic programs is widely recognized, particularly in temporal knowledge representation tasks [7].

7 CONCLUSIONS

We introduced a temporal deontic extension of Equilibrium Logic, characterizing achievement and maintenance obligations using repeaters. We illustrated how the most relevant challenges in temporal deontic reasoning can be accommodated in the new formalism. As fulfillment and violation of norms are concepts associated with an agent's actions, a natural step for future work is to extend the formalism to a multi-agent setting, for instance. We also plan to implement our formalism using temporal ASP tools, such as the TEL solver `telingo` [22], with potential applications including extending the monitoring framework introduced in [26, 48] to monitor normative properties.

ACKNOWLEDGMENTS

Work funded by (i) EU: grant No 10.1034440, (ii) WWTF: projects 10.47379/ICT22-023 and 10.47379/ICT23030, and (iii) the Spanish Ministry of Science, Innovation and Universities: grants PID2020-116201GB-I00 and PID2023-148531NB-I00.

REFERENCES

[1] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael Wooldridge. 2009. *A Temporal Logic of Normative Systems.* Springer Netherlands, Dordrecht, 69–106. https://doi.org/10.1007/978-1-4020-9084-4_5

[2] Felicidad Aguado, Pedro Cabalar, Martín Diéguez, Gilberto Pérez, Torsten Schaub, Anna Schuhmann, and Concepción Vidal. 2023. Linear-Time Temporal Answer Set Programming. *Theory and Practice of Logic Programming* 23, 1 (2023), 2–56.

[3] Felicidad Aguado, Pedro Cabalar, Jorge Fandinno, Gilberto Pérez, and Concepción Vidal. 2020. Explicit negation in linear-dynamic equilibrium logic. In *ECAI 2020*. IOS Press, 569–576.

[4] Felicidad Aguado, Pedro Cabalar, Gilberto Pérez, and Concepción Vidal. 2011. Loop Formulas for Splitable Temporal Logic Programs. In *LPNMR 2011 (LNCS, Vol. 6645)*. Springer, 80–92.

[5] Natasha Alechina, Giuseppe De Giacomo, Brian Logan, and Giuseppe Perelli. 2022. Automatic Synthesis of Dynamic Norms for Multi-Agent Systems. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, Vol. 19. 12–21.

[6] Marc Allaire and Guido Governatori. 2014. On the equivalence of defeasible deontic logic and temporal defeasible logic. In *PRIMA 2014*. Springer, 74–90.

[7] Krzysztof R Apt and Marc Bezem. 1991. Acyclic programs. *New generation computing* 9 (1991), 335–363.

[8] Shaun Azzopardi, Gordon J. Pace, Fernando Schapachnik, and Gerardo Schneider. 2021. On the Specification and Monitoring of Timed Normative Systems. In *RV 2021 (LNCS, Vol. 12974)*. Springer, 81–99.

[9] Blai Bonet and Hector Geffner. 2000. Planning with incomplete information as heuristic search in belief space. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*. 52–61.

[10] Laura Bozzelli and David Pearce. 2015. On the complexity of temporal equilibrium logic. In *ACM/IEEE LICS*. IEEE, 645–656.

[11] Laura Bozzelli and David Pearce. 2016. On the expressiveness of temporal equilibrium logic. In *ECAI*. Springer, 159–173.

[12] Gerhard Brewka, Thomas Eiter, and Miroslaw Truszczynski. 2011. Answer set programming at a glance. *Commun. ACM* 54, 12 (2011), 92–103.

[13] Gerd Brewka, Thomas Eiter, and Miroslaw Truszczyński (Eds.). 2016. *AI Magazine: special issue on Answer Set Programming*. AAAI Press. Volume 37, number 3. Editorial pp. 5-6.

[14] Jan Broersen and Julien Brunel. 2007. 'What I fail to do today, I have to do tomorrow': a logical study of the propagation of obligations. In *International Workshop on Computational Logic in Multi-Agent Systems*. Springer, 82–99.

[15] Jan Broersen, Andreas Herzig, and Nicolas Troquard. 2006. A stit-extension of ATL. In *European Workshop on Logics in Artificial Intelligence*. Springer, 69–81.

[16] Jan M. Broersen and Leendert W. N. van der Torre. 2011. Ten Problems of Deontic Logic and Normative Reasoning in Computer Science. In *ESSLLI 2010 (LNCS, Vol. 7388)*. Springer, 55–88.

[17] Pedro Cabalar. 2010. A normal form for linear temporal equilibrium logic. In *European Workshop on Logics in Artificial Intelligence*. Springer, 64–76.

[18] Pedro Cabalar, Agata Ciabattoni, and Leendert van der Torre. 2023. Deontic Equilibrium Logic with eXplicit Negation. In *JELIA 2023 (LNCS, Vol. 14281)*. Springer, 498–514.

[19] Pedro Cabalar and Stéphane Demri. 2011. Automata-based computation of temporal equilibrium models. In *International Symposium on Logic-Based Program Synthesis and Transformation*. Springer, 57–72.

[20] Pedro Cabalar, Thomas Eiter, and Davide Soldà. 2024. Contracted Temporal Equilibrium Logic. In *Proceedings of the 21th International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam, November 2-8, 2024*.

[21] Pedro Cabalar, Jorge Fandinno, and Luis Fariñas del Cerro. 2020. Autoepistemic Answer Set Programming. *Artificial Intelligence* 289 (2020), 103382.

[22] Pedro Cabalar, Roland Kaminski, Philip Morkisch, and Torsten Schaub. 2019. telingo = ASP + Time. In *LPNMR 2019 (LNCS, Vol. 11481)*. Springer, 256–269.

[23] José Carmo and Andrew JI Jones. 2002. Deontic logic and contrary-to-duties. In *Handbook of Philosophical Logic: Volume 8*. Springer, 265–343.

[24] Roderick Chisholm. 1963. Contrary-to-duty imperatives and deontic logic. *Analysis* 24, 2 (1963), 33–36.

[25] Thomas Eiter, Wolfgang Faber, Nicola Leone, Gerald Pfeifer, and Axel Polleres. 2000. Planning under incomplete knowledge. In *International Conference on Computational Logic*. Springer, 807–821.

[26] Thomas Eiter and Davide Soldà. 2024. Computational Aspects of Progression for Temporal Equilibrium Logic. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3342–3350. https://doi.org/10.24963/ijcai.2024/370 Main Track.

[27] François Fages. 1994. Consistency of Clark's completion and existence of stable models. *Methods Log. Comput. Sci.* 1, 1 (1994), 51–60.

[28] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. 2006. A generalization of the Lin-Zhao theorem. *Annals of Mathematics and Artificial Intelligence* 47, 1-2 (2006), 79–101.

[29] James William Forrester. 1984. Gentle murder, or the adverbial Samaritan. *The Journal of Philosophy* 81, 4 (1984), 193–197.

[30] M. Gelfond and V. Lifschitz. 1991. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing* 9, 3–4 (1991), 365–386.

[31] Laura Giordano, Alberto Martelli, and Daniele Theseider Dupré. 2013. Reasoning about actions with temporal answer sets. *Theory and Practice of Logic Programming* 13, 2 (2013), 201–225.

[32] Laura Giordano, Alberto Martelli, and Daniele Theseider Dupré. 2013. Temporal deontic action logic for the verification of compliance to norms in ASP. In *ICAIL 2013*. ACM, 53–62.

[33] Ricardo Gonçalves and José Júlio Alferes. 2013. Deontic logic programs. In *AAMAS*. IFAAMAS, 1333–1334.

[34] Guido Governatori. 2024. An ASP Implementation of Defeasible Deontic Logic. *KI-Künstliche Intelligenz* (2024), 1–10.

[35] Guido Governatori, Joris Hulstijn, Régis Riveret, and Antonino Rotolo. 2007. Characterising deadlines in temporal modal defeasible logic. In *Advances in Artificial Intelligence*. Springer, 486–496.

[36] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Simone Scannapieco. 2013. Computing strong and weak permissions in defeasible logic. *Journal of Phil. Logic* 42, 6 (2013), 799–829.

[37] Guido Governatori and Antonino Rotolo. 2013. Computing temporal defeasible logic. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*. Springer, 114–128.

[38] Christian Hatschka, Agata Ciabattoni, and Thomas Eiter. 2023. Deontic Paradoxes in ASP with Weak Constraints. In *Proceedings of ICLP 2023 (EPTCS, Vol. 385)*. 367–380.

[39] John Horty. 2001. *Deontic logic and Agency*. OUP.

[40] Wojciech Jamroga, Wiebe van der Hoek, and Michael Wooldridge. 2004. On obligations and abilities. In *International Workshop on Deontic Logic in Computer Science*. Springer, 165–181.

[41] Robert A. Kowalski and Ken Satoh. 2018. Obligation as Optimal Goal Satisfaction. *J. Phil. Log.* 47, 4 (2018), 579–609.

[42] Vladimir Lifschitz, David Pearce, and Agustin Valverde. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL)* 2, 4 (2001), 526–541.

[43] Zohar Manna and Amir Pnueli. 1992. *The temporal logic of reactive and concurrent systems - specification*. Springer.

[44] Donald Nute. 1997. *Defeasible deontic logic*. Vol. 263. Springer Science & Business Media.

[45] David Pearce. 2006. Equilibrium logic. *Ann. Math. Artif. Intell.* 47, 1-2 (2006), 3–41.

[46] Alf Ross. 1944. Imperatives and logic. *Philosophy of Science* 11, 1 (1944), 30–46.

[47] John Rogers Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, England: Cambridge University Press.

[48] Davide Soldà, Ignacio D. Lopez-Miguel, Ezio Bartocci, and Thomas Eiter. 2023. Progression for Monitoring in Temporal ASP. In *ECAI 2023*, Vol. 372. IOS Press, 2170–2177.

[49] R. Thomason. 1981. Deontic logic as founded on tense logic. In *New Studies in Deontic Logic: Norms, Actions, and the Foundation of Ethics*. Reidel Publishing Company.

[50] G. Tseitin. 1968. On the complexity of derivation in the propositional calculus. *Zapiski nauchnykh seminarov LOMI* 8 (1968), 234–259.

[51] Georg Henrik von Wright. 1967. Deontic Logics. *American Philosophical Quarterly* 4, 2 (1967), 136–143.