Towards Action Languages with Norms and Deadlines

Matthias Knorr **Alfredo Gabaldon** **Ricardo Gonçalves** **João Leite** **Martin Slota**

CENTRIA & Departamento de Informática
Universidade Nova de Lisboa
2829-516 Caparica, Portugal

Abstract

Action Languages are simple logical formalisms to describe the properties of a domain and the behavior of an agent and to reason about it. They offer an elegant solution to the frame problem, but are inapt to reason with norms in which an obligation deadline may require the agent to adapt its behavior even though no action occurred. In this paper we extend the Action Language $\mathcal{A}$ with features that allow reasoning about norms and time in dynamic domains. Unlike previous extensions of Action Languages with norms, our resulting language is expressive enough to handle different kinds of obligations with deadlines that explicitly refer to time, as well as norm violations and contrary-to-duty obligations.

1 Introduction

Open dynamic systems, e.g., systems interacting on the Web, social systems and open agent communities, have attracted increased attention in recent years. In these systems, constraints on the behavior of the participants cannot be hardwired in their specification. Instead, desirable properties are promoted by *normative systems* [Esteva *et al.*, 2001; Boella and van der Torre, 2004; Boella *et al.*, 2008; Alechina *et al.*, 2012; Bulling and Dastani, 2011; Sadiq *et al.*, 2007]. Norms, when used to govern *autonomous* agents, do not simply act as hard constraints that prevent the agent from adopting some behavior, but rather provide an indication as to how the agent *should* behave which, if not adhered to, can result in the application of sanctions or other normative effects.

In general, *norms* can be seen as a specification of what is expected to follow from a specific state of affairs, e.g., in the form of *obligations*. There are *obligations to-do*, i.e., obligations to execute an action before a deadline—e.g., to reply within one day after receiving a request, or to register before logging in; *obligations to-achieve*, i.e., obligations to bring about, before the deadline, a state of the world in which some proposition holds—e.g., to achieve a certain amount of credits within the academic year; and *obligations to-maintain*, i.e., obligations to maintain a state of the world in which some proposition holds until the deadline—e.g., to keep the contract with your mobile phone company for one year.

One important characteristic of realistic systems of norms is the prominent role of *time* and *deadlines*. Another feature of complex systems of norms are *reparative obligations*, or *contrary-to-duty obligations* [Carmo and Jones, 2002], i.e., obligations imposed as a consequence of the violation of some other obligation—e.g., to pay a fine within 10 days if the obligation to return an item by a deadline is violated.

Action Languages [Gelfond and Lifschitz, 1998] are simple logical formalisms for modeling dynamic systems, with application in *Artificial Intelligence*, *Robotics* and *Multi-Agent Systems*. A theory of an *Action Language* describes the properties of a domain and the abilities of an agent, compactly specifying a transition diagram containing all possible trajectories of the system. Action Languages provide a simple and elegant solution to the *frame problem* and enable an agent to encode the applicability of actions, their effects, describe complex interrelations between fluents and use automated planning to achieve some goal.

The combination of Action Languages and norms has received some attention in the literature. The line of work in [Craven and Sergot, 2008; Artikis *et al.*, 2009] extends action language $\mathcal{C}+$ [Giunchiglia *et al.*, 2004] for representing norms and institutional aspects of normative societies, focusing on aspects such as power and count-as rules. In [Gelfond and Lobo, 2008], an action language is extended with propositions for specifying defeasible authorization and obligation policies, but only obligations to-do are considered. However, none of the previous approaches deals with explicit time, deadlines or contrary-to-duty obligations. As it turns out, existing Action Languages, and these extensions, cannot capture the dynamics of deadlines because of the fundamental role they assign to physical *actions*, whose execution is *the only way* to cause a state change. With the introduction of norms with deadlines, and contrary-to-duty obligations, state change needs to also be triggered by the violation of other obligations, resulting from the expiration of the deadlines, which cannot be encoded in existing Action Languages.

To address this limitation, in this paper, we extend the Action Language $\mathcal{A}$[1] [Gelfond and Lifschitz, 1998] with features that allow reasoning about norms and time in dynamic do-

[1]We restrict ourselves to $\mathcal{A}$ and focus on explaining the technical details related to norms with explicit time deadlines, leaving more expressive Action Languages for future work.

mains, resulting in the Normative Action Language $\mathcal{A}_\mathcal{N}$ and a query language that can deal with

- *obligations to-do*, *to-achieve* and *to-maintain*;
- deadlines that explicitly refer to time;
- norm violations and satisfactions;
- contrary-to-duty obligations.

At the same time, our approach solves the frame problem also for obligations and it is more amenable to implementation than other related work based on more complex formalisms (see related work in Sect. 4).

After introducing, in Sect. 2, the syntax and semantics of our normative Action Language $\mathcal{A}_\mathcal{N}$, illustrating its use, and presenting some basic properties, in Sect. 3 we present a query language for $\mathcal{A}_\mathcal{N}$, discuss its complexity and equivalence between theories in $\mathcal{A}_\mathcal{N}$, before we conclude in Sect. 4.[2]

2 Normative Action Language $\mathcal{A}_\mathcal{N}$

We introduce the syntax and semantics of $\mathcal{A}_\mathcal{N}$, a simple language for specifying norms, yet expressive enough to handle different kinds of obligations with deadlines, their satisfaction and violation, and contrary-to-duty obligations. We start from the deterministic Action Language $\mathcal{A}$ [Gelfond and Lifschitz, 1998] whose semantics builds on transition systems in which nodes correspond to states of the environment and edges correspond to transitions between states and are labeled by the action that causes the transition. To capture the meaning of a set of norms, we extend this transition system by expanding the states with a deontic component, and by adding a temporal dimension to transitions.

2.1 Syntax

Action Languages provide two disjoint, non-empty sets of function-free first-order atoms[3] defined over a given signature $\Sigma = \langle \mathcal{P}, \mathcal{C}, \mathcal{V} \rangle$ of pairwise disjoint sets of predicates ($\mathcal{P}$), constants ($\mathcal{C}$) and variables ($\mathcal{V}$): a set $\mathcal{A}$ of *elementary actions* and a set $\mathcal{F}$ of *physical fluents*. An *action* is a finite, possibly empty subset of $\mathcal{A}$ and can be understood as a set of elementary actions that are executed simultaneously. If convenient, we denote a singleton action $\{ \alpha \}$ with the elementary action α. Physical fluents $f \in \mathcal{F}$ and their negations $\neg f$ form the set of *physical literals*, used to represent states of the "world."

To allow for deontic expressions, we extend the signature Σ with sets of *time points* $\mathcal{T}$ and *time variables* $\mathcal{V}_t$, resulting in the deontic signature $\Sigma_d = \langle \mathcal{P}, \mathcal{C} \cup \mathcal{T}, \mathcal{V} \cup \mathcal{V}_t \rangle$. From now on, we assume an arbitrary but fixed deontic signature Σ_d.

Both additions to the signature are related to time, and we explain them in the following. The set of time points $\mathcal{T}$ represents the time domain, and we assume that $\mathcal{T}$ is a countable subset of non-negative real numbers, including 0, such as natural numbers $\mathbb{N}$. The set of time variables $\mathcal{V}_t$ relates specifically to $\mathcal{T}$ in the same standard way as $\mathcal{V}$ relates to $\mathcal{C}$, and we reserve a special time variable $\mathbf{now} \in \mathcal{V}_t$ which we always associate with the time point representing the current time.

Both $\mathcal{T}$ and $\mathcal{V}_t$ are used to define *time expressions*, which allow us to shift time points into the future by a specific time interval. The set of *time expressions* $\mathcal{T}^*$ is defined as $\mathcal{T}^* = \mathcal{T} \cup \{ V + c \mid V \in \mathcal{V}_t \wedge c \in \mathcal{T} \}$.[4] Where convenient, we simply abbreviate $V + 0$ by V.

We now introduce *deontic literals* to represent three types of obligations: 1. *obligations to-do*, requiring that an action be executed; 2. *obligations to-achieve*, requiring that a physical literal become true; 3. *obligations to-maintain*, requiring that a physical literal remain true; all three strictly before a specified deadline.[5]

Definition 1 (Deontic literal)**.** Let $A \subseteq \mathcal{A}$ be a non-empty action, l a physical literal, and $t \in \mathcal{T}^*$, called the *deadline*. An *obligation* is of the following three forms:

- *obligation to-do* $\mathsf{O}^{\mathsf{d}}_t\, A$;
- *obligation to-achieve* $\mathsf{O}^{\mathsf{a}}_t\, l$;
- *obligation to-maintain* $\mathsf{O}^{\mathsf{m}}_t\, l$.

Obligations and their negations form the set of *deontic literals*. The expression $\mathsf{O}_t\, l$ represents both $\mathsf{O}^{\mathsf{a}}_t\, l$ and $\mathsf{O}^{\mathsf{m}}_t\, l$.

Note that obligations to-achieve and to-maintain can be understood as dual: the intention for the former is to require that literal l holds for (at least) one time point strictly in between the introduction of the obligation and its deadline, and for the latter that l holds for all time points from the time point of the introduction until (immediately before) the deadline. This will be accordingly reflected in the semantics later.

A *literal* is either a physical literal or a deontic literal. A literal is *ground* if it contains no variables. Literals f and $\neg f$ are called *complementary*. The literal complementary to l is denoted by $\bar{l}$.

In Action Languages, only actions can cause state changes, but the introduction of obligations with deadlines should allow a state change to be triggered also by the violation (V) or the satisfaction (S) of obligations, resulting from the expiration of their deadlines. Deontic events accommodate for that.

Definition 2 (Event)**.** Let d be an obligation. *Deontic events* are expressions of the form $\mathsf{V}d$ and $\mathsf{S}d$. An *event* is an action or a deontic event.

We recall propositions in $\mathcal{A}$ and at the same time extend them to include norms.

Definition 3 (Norm and normative specification)**.** Let e be an event, l a deontic or physical literal and, for all i with $1 \leq i \leq$

[2] An extended version of the paper containing proofs can be found at `http://centria.di.fct.unl.pt/~mknorr/NRAC13.pdf`.

[3] In [Gelfond and Lifschitz, 1998], only the propositional case is considered. We use function-free first-order atoms here to ease the presentation of our formalization of time.

[4] Here we abuse the set of time points and time variables to also represent *time intervals*. The expression $V + c$ always represents the addition of a time interval to a time point, or of two time intervals.

[5] The restriction to non-inclusive deadlines is an arbitrary decision and it would be reasonable to consider inclusive deadlines instead, or even to introduce both types of deadlines. For simplicity, we consider only non-inclusive deadlines.

n, l_i literals. A *proposition* n takes the following form:

$$e \textbf{ causes } l \textbf{ if } l_1, \ldots, l_n \ . \tag{1}$$

We say that e is the *event* of n, l the *effect* of n and $l_1, \ldots, l_n$ its *condition*. If the condition is empty, we write (e **causes** l). If l is a deontic literal, then n is a *norm*. If l is a physical literal, then e is an action, and all l_i in the condition are physical literals.

A proposition is *safe* if every variable (different from $\mathbf{now}$) appearing in its effect also appears in its event or in an obligation within its condition. A *normative specification* $\mathcal{N}$ is a finite set of safe propositions of the form (1).

Intuitively, a norm of the form (1) adds or removes the obligation specified by l if the *event* occurs and the *condition* is satisfied. Also note that a proposition with physical literal l matches a proposition in $\mathcal{A}$ [Gelfond and Lifschitz, 1998] and the rationale for the applied restrictions is that normative information should not affect the physical world. This is indeed the case and in line with the idea that obligations are meant to represent only guidelines of desired behavior for an agent (including penalties for non-compliance), unlike the line of work in which obligations can be used to prohibit the execution of an action (see, e.g., [Cholvy, 1999]). Finally, safeness of variables occurring in l prevents from the specification of propositions with non-ground effects.[6]

Example 4. Consider a set of norms in a university library scenario:

$$\begin{aligned}
&\mathit{borrow}(X) \textbf{ causes } \mathsf{O}^{\mathsf{d}}_{\mathbf{now}+4}\, \mathit{ret}(X) \textbf{ if } \mathit{ugrad} && (2)\\
&\mathit{borrow}(X) \textbf{ causes } \mathsf{O}^{\mathsf{d}}_{\mathbf{now}+12}\, \mathit{ret}(X) \textbf{ if } \mathit{grad} && (3)\\
&\mathit{renew}(X) \textbf{ causes } \mathsf{O}^{\mathsf{d}}_{T+4}\, \mathit{ret}(X) \textbf{ if } \mathsf{O}^{\mathsf{d}}_{T}\, \mathit{ret}(X) && (4)\\
&\mathit{renew}(X) \textbf{ causes } \neg\mathsf{O}^{\mathsf{d}}_{T}\, \mathit{ret}(X) \textbf{ if } \mathsf{O}^{\mathsf{d}}_{T}\, \mathit{ret}(X) && (5)\\
&\mathsf{VO}^{\mathsf{d}}_{T}\, \mathit{ret}(X) \textbf{ causes } \mathsf{O}^{\mathsf{d}}_{\mathbf{now}+1}\, \mathit{pay} && (6)\\
&\mathsf{VO}^{\mathsf{d}}_{T}\, \mathit{ret}(X) \textbf{ causes } \mathsf{O}^{\mathsf{d}}_{\mathbf{now}+1}\, \mathit{ret}(X) \textbf{ if } \mathit{ugrad} && (7)\\
&\mathsf{VO}^{\mathsf{d}}_{T}\, \mathit{ret}(X) \textbf{ causes } \mathsf{O}^{\mathsf{d}}_{\mathbf{now}+3}\, \mathit{ret}(X) \textbf{ if } \mathit{grad} && (8)
\end{aligned}$$

Norms (2) and (3) specify that borrowing a book creates the obligation to return that book within the period specified depending on the student's status (4 and 12 weeks for undergraduate and graduate students respectively). A book may be renewed for 4 more weeks, which means updating the obligation with the new deadline (4–5). Finally, a contrary-to-duty norm specifies that, if a user fails to return the book on time, a fine has to be paid within one week (6) and the book has to be returned (7–8).

On different domains, an example of a norm with an achievement obligation is that one has the obligation to achieve 30 credits within the academic year, and an example of a norm with a maintenance obligation is that one has the obligation to maintain a contract with a mobile carrier for (at least) 24 months.

$$\mathit{enterAcademicYear} \textbf{ causes } \mathsf{O}^{\mathsf{a}}_{\mathbf{now}+12}\, \mathit{sumCredits}(\mathit{30})$$

$$\mathit{startMobileContract} \textbf{ causes } \mathsf{O}^{\mathsf{m}}_{\mathbf{now}+24}\, \mathit{mobileContract}$$

[6] A less restrictive condition could be applied to propositions whose effect is a physical literal, but this would only affect the action language which is not our major concern.

2.2 Semantics

The semantics of Action Language $\mathcal{A}$ is defined as a transition system T modeling the physical environment. A node σ of T represents a possible physical state and a transition $\langle \sigma, A, \sigma' \rangle$ represents that state σ' can be reached from σ by executing action A. We extend such T with deontic features and time to define the semantics of normative specifications as follows. We augment states σ with deontic states δ and we define when literals are satisfied in such a combined state σ/δ. Next, we present a relation that captures how a deontic event is caused by either reaching a deadline or due to an executed action. We proceed by specifying which positive and negative normative effects are triggered in a state σ/δ when executing an action A at time t w.r.t. $\mathcal{N}$, using a Boolean function $\rho_{A,t}()$ in the former case to avoid introducing meaningless obligations. This enables us to define a resulting new deontic state and, subsequently, transitions and paths in the resulting transition system $T_{\mathcal{N}}$.

Let $\mathcal{N}$ be a normative specification. The states of the transition system $T_{\mathcal{N}}$ consist of two parts: a set of physical literals representing the physical "world," and a set of obligations representing the deontic state of the agent. Additionally, we require that obligations be part of the state only if they are not immediately satisfied or violated.

Definition 5 (State of $T_{\mathcal{N}}$)**.** Let σ be a complete and consistent set of ground physical literals, i.e., for each ground physical fluent f, exactly one of f and $\neg f$ belongs to σ, and δ a finite set of ground obligations. Then, σ/δ is a *state* of the transition system $T_{\mathcal{N}}$ if the following conditions are satisfied for every physical literal l and deadline t: ($\mathsf{O}^{\mathsf{a}}_{t}\, l \notin \delta$ or $l \notin \sigma$) and ($\mathsf{O}^{\mathsf{m}}_{t}\, l \notin \delta$ or $l \in \sigma$). We call σ the *physical state* and δ the *deontic state*.

Note that, unlike σ, δ is not complete, since it would be impractical to require that d or $\neg d$ occur in δ for each d due to the usually infinite set of time points $\mathcal{T}$. This is also why we consider a separate set δ and do not merge the two parts into one.

To deal with the satisfaction of non-ground literals in a state σ/δ, we introduce a *variable assignment* z as a function mapping variables to constants ($\mathcal{V} \rightarrow \mathcal{C}$) and time variables to time points ($\mathcal{V}_t \rightarrow \mathcal{T}$). For every time point t, we denote the set of variable assignments z such that $\mathsf{z}(\mathbf{now}) = t$ by $\mathcal{Z}_t$. Hence, the index t in $\mathcal{Z}_t$ is not merely notation, but defines the value that is assigned to $\mathbf{now}$.

For any literal or event λ, we denote by $\lambda|_{\mathsf{z}}$ the literal or event obtained from λ by substituting every variable according to z, and, subsequently, replacing every time expression $t + c$ with the time point t' such that $t' = t + c$. E.g., $\mathsf{O}^{\mathsf{d}}_{9}\, \mathit{ret}(\mathit{book})$ is the result of $\left(\mathsf{O}^{\mathsf{d}}_{\mathbf{now}+4}\, \mathit{ret}(X)\right)\big|_{\{\, X \rightarrow \mathit{book},\, \mathbf{now} \rightarrow 5 \,\}}$.

Satisfaction for ground literals in a state σ/δ is defined as follows for a physical literal l and a ground obligation d:

$$\begin{aligned}
\sigma/\delta &\models l \text{ iff } l \in \sigma \ ,\\
\sigma/\delta &\models d \text{ iff } d \in \delta \ ,\\
\sigma/\delta &\models \neg d \text{ iff } d \notin \delta \ .
\end{aligned}$$

Furthermore, given a variable assignment z, and a set of literals $L = \{l_1, \dots, l_n\}$, we define $\sigma/\delta \models L|_{\mathsf{z}}$ iff for all i with $i \in \{1, \dots, n\}$, $\sigma/\delta \models l_i|_{\mathsf{z}}$.

Each transition of $T_{\mathcal{N}}$ is a tuple $\langle \sigma/\delta, (A, t), \sigma'/\delta' \rangle$, where A is a ground action and $t \in \mathcal{T}$ a time point, meaning that A occurred at time t, causing the transition from state σ/δ to σ'/δ'. Since the physical effects are independent of the deontic ones, we first define a relation $R_{\mathcal{N}}$ that, for a given $\mathcal{N}$, associates each physical state σ and ground action A with a new physical state σ':

$$\langle \sigma, A, \sigma' \rangle \in R_{\mathcal{N}} \text{ iff } \sigma' = (\sigma \cup \mathsf{E}_A(\sigma)) \setminus \{\, \bar{l} \mid l \in \mathsf{E}_A(\sigma) \,\} \quad ,$$

where $\mathsf{E}_A(\sigma)$ stands for the set of all physical literals $l|_{\mathsf{z}}$ such that $(e$ **causes** l **if** $C) \in \mathcal{N}$ and there is $\mathsf{z} \in \mathcal{Z}_t$ with $\sigma/\delta \models C|_{\mathsf{z}}$ and $e|_{\mathsf{z}} \subseteq A$. If $A = \emptyset$, then $\sigma' = \sigma$, which allows us to handle deontic updates resulting from deadline expirations at time points in which no action occurs. Note that the requirement that σ' be a physical state ensures that $\langle \sigma, A, \sigma' \rangle \notin R_{\mathcal{N}}$ if A has contradictory effects in state σ.

We proceed by specifying how to obtain a new deontic state. First, we define the conditions for the occurrence of deontic events, which are satisfactions/violations of obligations occurring in the current deontic state δ w.r.t. the new physical state σ'.

Definition 6 (Occurrence of deontic event)**.** Let σ/δ be a state of $T_{\mathcal{N}}$, A, B ground actions, $\langle \sigma, A, \sigma' \rangle \in R_{\mathcal{N}}$ and t, t' time points. The occurrence relation for ground deontic events under action A at time t, $\vdash_{A,t}$, is defined for tuples $\langle \delta, \sigma' \rangle$ as follows:

$$\begin{array}{lll}
\langle \delta, \sigma' \rangle \vdash_{A,t} \mathsf{VO}^{\mathsf{d}}_{t'} B & \text{iff} & \mathsf{O}^{\mathsf{d}}_{t'} B \in \delta \wedge t \geq t' \\
\langle \delta, \sigma' \rangle \vdash_{A,t} \mathsf{VO}^{\mathsf{a}}_{t'} l & \text{iff} & \mathsf{O}^{\mathsf{a}}_{t'} l \in \delta \wedge t \geq t' \\
\langle \delta, \sigma' \rangle \vdash_{A,t} \mathsf{SO}^{\mathsf{m}}_{t'} l & \text{iff} & \mathsf{O}^{\mathsf{m}}_{t'} l \in \delta \wedge t \geq t' \\
\langle \delta, \sigma' \rangle \vdash_{A,t} \mathsf{SO}^{\mathsf{d}}_{t'} B & \text{iff} & \mathsf{O}^{\mathsf{d}}_{t'} B \in \delta \wedge t < t' \wedge B \subseteq A \\
\langle \delta, \sigma' \rangle \vdash_{A,t} \mathsf{SO}^{\mathsf{a}}_{t'} l & \text{iff} & \mathsf{O}^{\mathsf{a}}_{t'} l \in \delta \wedge t < t' \wedge l \in \sigma' \\
\langle \delta, \sigma' \rangle \vdash_{A,t} \mathsf{VO}^{\mathsf{m}}_{t'} l & \text{iff} & \mathsf{O}^{\mathsf{m}}_{t'} l \in \delta \wedge t < t' \wedge \bar{l} \in \sigma'
\end{array}$$

Additionally, $\epsilon_{A,t}(\delta, \sigma') = \{\, e \mid \langle \delta, \sigma' \rangle \vdash_{A,t} e \,\}$.

The above conditions encode the dynamics of violations and satisfactions, and depend on the type of obligation involved. The first three represent events generated by a deadline expiration. The last three represent events that occur before the expiration of the respective deadline. Namely, either action B is executed (as part of A) at time t, or a state change affects the literal l to be achieved (or cease to be maintained). We explain the latter case in more detail for an obligation to-achieve l. Such an obligation can only be part of a state σ/δ if $\bar{l} \in \sigma$. If executing action A at time t introduces l, i.e., adds it to the new state σ' (and removes $\bar{l}$), then an event occurs, which (as we will see below) is used to trigger the removal of the corresponding obligation, but also possibly the introduction of new obligations.

Before defining the normative effects of executing action A at time t in state σ/δ, we need to introduce an auxiliary function $\rho_{A,t}(d, \sigma')$ that determines whether, given σ' with $\langle \sigma, A, \sigma' \rangle \in R_{\mathcal{N}}$, an obligation d, which would be introduced to the new deontic state, is *relevant*: $\rho_{A,t}(d, \sigma') = \bot$ if either (1) d is an obligation with deadline $t' \leq t$, (2) $d = \mathsf{O}^{\mathsf{d}}_{t'} B \wedge B \subseteq A$, (3) $d = \mathsf{O}^{\mathsf{a}}_{t'} l \wedge l \in \sigma'$, or (4) $d = \mathsf{O}^{\mathsf{m}}_{t'} l \wedge \bar{l} \in \sigma'$; otherwise $\rho_{A,t}(d, \sigma') = \top$. Condition (1) matches the first part of Def. 6, while (2-4) matches the second. We thus avoid the introduction of obligations that would be satisfied/violated immediately, following the rationale to only consider obligations whose satisfaction can be influenced by the agent's behavior.

We now define the normative effects of executing an action A at time t in a given state σ/δ. We say that an effect of a norm is positive (negative) if it is an obligation (its negation). For each instance of a norm in $\mathcal{N}$ we need to evaluate its condition in σ/δ, check whether the respective event is a subset of action A or a deontic event, and, in case of the positive effects, check if the effect of the norm is an obligation that is relevant (or can be safely ignored). The latter and the check for deontic events occur w.r.t. the new physical state σ' (obtained by executing A on σ) as already indicated.

Definition 7 (Normative effect)**.** Let σ/δ be a state of $T_{\mathcal{N}}$, $\langle \sigma, A, \sigma' \rangle \in R_{\mathcal{N}}$, t a time point and d an obligation. The set of *positive normative effects* $\mathsf{E}^+_{A,t}(\sigma/\delta, \sigma')$ and the set of *negative normative effects* $\mathsf{E}^-_{A,t}(\sigma/\delta, \sigma')$ are defined as follows:

$$\begin{aligned}
\mathsf{E}^+_{A,t}(\sigma/\delta, \sigma') = \{\, (d|_{\mathsf{z}}) \mid (e \textbf{ causes } d \textbf{ if } C) \in \mathcal{N} \wedge \exists \mathsf{z} \in \mathcal{Z}_t :& \\
\sigma/\delta \models C|_{\mathsf{z}} \wedge (e|_{\mathsf{z}} \subseteq A \vee \langle \delta, \sigma' \rangle \vdash_{A,t} e|_{\mathsf{z}})& \\
\wedge\, \rho_{A,t}(d|_{\mathsf{z}}, \sigma') \,\};& \\
\mathsf{E}^-_{A,t}(\sigma/\delta, \sigma') = \{\, (d|_{\mathsf{z}}) \mid (e \textbf{ causes } \neg d \textbf{ if } C) \in \mathcal{N} \wedge \exists \mathsf{z} \in \mathcal{Z}_t :& \\
\sigma/\delta \models C|_{\mathsf{z}} \wedge (e|_{\mathsf{z}} \subseteq A \vee \langle \delta, \sigma' \rangle \vdash_{A,t} e|_{\mathsf{z}}) \,\}.&
\end{aligned}$$

The new deontic state δ' can now be computed from σ/δ by first detecting which deontic events occur (and removing the corresponding obligations), then adding the positive effects of these events and finally removing their negative effects.

Definition 8 (New deontic state)**.** Let σ/δ be a state of $T_{\mathcal{N}}$, $\langle \sigma, A, \sigma' \rangle \in R_{\mathcal{N}}$, t a time point and d an obligation. We define $\mathsf{G}(\mathsf{V}d) = \mathsf{G}(\mathsf{S}d) = d$, for any set of deontic events E, $\mathsf{G}(E) = \{\, \mathsf{G}(e) \mid e \in E \,\}$ and the new deontic state

$$\delta' = \Big[(\delta \setminus \mathsf{G}(\epsilon_{A,t}(\delta, \sigma'))) \cup \mathsf{E}^+_{A,t}(\sigma/\delta, \sigma')\Big] \setminus \mathsf{E}^-_{A,t}(\sigma/\delta, \sigma').$$

Three consequences follow immediately: first, if an obligation is introduced and removed simultaneously by different norms, then the removal prevails, following a generalization of the *in dubio pro reo* principle; second, it may happen that the occurrence of a deontic event removes some obligation, which is immediately re-introduced in $\mathsf{E}^+_{A,t}()$ if a corresponding norm exists, such as for example if you pay a fine and, at the same time, commit an offense that incurs in the same penalty; and third, the frame problem for obligations is trivially solved in this equation—whatever appears in δ and is not removed on purpose, persists in δ'.

We show that σ' and δ' indeed form a state of $T_{\mathcal{N}}$.

Proposition 9. *Let σ/δ be a state of $T_{\mathcal{N}}$, $\langle \sigma, A, \sigma' \rangle \in R_{\mathcal{N}}$, and δ' as defined in Def. 8. Then σ'/δ' is a state of $T_{\mathcal{N}}$.*

Furthermore, considering the definition of deontic events, whenever a deadline of an existing obligation is reached, a deontic event always takes place. A consequence of this observation is that a transition from σ/δ must not occur at a time point that exceeds the deadline of some obligation in δ. We define this time point as the earliest deadline among the current obligations, or infinity if there are no obligations in δ. Formally, let $d(\delta) = \{\, t \in \mathcal{T} \mid \mathsf{O}_t\, l \in \delta \text{ or } \mathsf{O}^{\mathsf{d}}_t\, B \in \delta \,\}$. Then, $\mathsf{ltp}(\delta) = \min(d(\delta))$ if $d(\delta) \neq \emptyset$ and $\mathsf{ltp}(\delta) = \infty$ if $d(\delta) = \emptyset$. Note that, since δ is assumed finite, this notion of least time point is well-defined, i.e., if $d(\delta) \neq \emptyset$, then $\mathsf{ltp}(\delta) \in d(\delta)$, which, along with Proposition 9, allows us to define transitions of $T_{\mathcal{N}}$:

Definition 10 (Transition)**.** A *transition of* $T_{\mathcal{N}}$ is a tuple $\langle \sigma/\delta, (A,t), \sigma'/\delta' \rangle$ where σ, δ, σ', δ' and A are as in Def. 8, t is a time point s.t. $t \leq \mathsf{ltp}(\delta)$ and if $A = \emptyset$, then $t = \mathsf{ltp}(\delta)$.

Example 11. The following are transitions of $T_{\mathcal{N}}$ for Example 4 in Sect. 2.1.

$$\langle \{ugrad\}/\emptyset, (borrow(b), 1), \{ugrad\}/\{\mathsf{O}^{\mathsf{d}}_5\, ret(b)\}\rangle$$
$$\langle \{ugrad\}/\{\mathsf{O}^{\mathsf{d}}_5\, ret(b)\}, (ret(b), 4), \{ugrad\}/\emptyset\rangle$$
$$\langle \{ugrad\}/\{\mathsf{O}^{\mathsf{d}}_5\, ret(b)\}, (\emptyset, 5), \{ugrad\}/\{\mathsf{O}^{\mathsf{d}}_6\, pay, \mathsf{O}^{\mathsf{d}}_6\, ret(b)\}\rangle.$$

The tuple $\langle \{ugrad\}/\{\mathsf{O}^{\mathsf{d}}_5\, ret(b)\}, (ret(b), 8), \{ugrad\}/\emptyset\rangle$ is not a transition because $\mathsf{ltp}(\{\mathsf{O}^{\mathsf{d}}_5\, ret(b)\}) = 5 \not\geq 8$.

We can show that the transition system $T_{\mathcal{N}}$ is deterministic.

Proposition 12. *$T_{\mathcal{N}}$ is deterministic, i.e., if $\langle \sigma/\delta, (A,t), \sigma'/\delta' \rangle$ and $\langle \sigma/\delta, (A,t), \sigma''/\delta'' \rangle$ are transitions of $T_{\mathcal{N}}$, then $\sigma'/\delta' = \sigma''/\delta''$.*

Now, a path is an alternating sequence of states in $T_{\mathcal{N}}$ and pairs (A,t) corresponding to the transitions of $T_{\mathcal{N}}$.

Definition 13 (Path)**.** A *path* is a sequence of the form

$$\sigma_0/\delta_0, (A_1,t_1), \sigma_1/\delta_1, \ldots, (A_n,t_n), \sigma_n/\delta_n \quad , \qquad (9)$$

where σ_j/δ_j is a state of $T_{\mathcal{N}}$ for every $0 \leq j \leq n$, $\langle \sigma_j/\delta_j, (A_{j+1}, t_{j+1}), \sigma_{j+1}/\delta_{j+1} \rangle$ is a transition of $T_{\mathcal{N}}$ for every $0 \leq j < n$, and $t_j < t_{j+1}$ for every $1 \leq j < n$.

The last condition states the assumption that the time points in a path are ordered.

The satisfaction of an obligation to-do or to-achieve and the violation of an obligation to-maintain always indicate some relevant change w.r.t. the previous state.

Proposition 14. *Let P be a path of the form (9).*

If $\langle \delta_{j-1}, \sigma_j \rangle \vdash_{A_j, t_j} \mathsf{SO}^{\mathsf{d}}_t\, B$, then $B \not\subseteq A_{j-1}$ and $B \subseteq A_j$;
if $\langle \delta_{j-1}, \sigma_j \rangle \vdash_{A_j, t_j} \mathsf{SO}^{\mathsf{a}}_t\, l$, then $l \notin \sigma_{j-1}$ and $l \in \sigma_j$;
if $\langle \delta_{j-1}, \sigma_j \rangle \vdash_{A_j, t_j} \mathsf{VO}^{\mathsf{m}}_t\, l$, then $l \in \sigma_{j-1}$ and $l \notin \sigma_j$.

A symmetric result for the other three deontic events does not hold, simply because these occur due to a deadline that is reached with the progress of time.

3 Query Language and Equivalence

We now define a query language for $\mathcal{A}_{\mathcal{N}}$ that can be used to check whether a certain literal/event occurs in a specific time interval given a normative specification and a description of the initial state. We consider decidability and complexity of answering queries. Then, we also discuss equivalence between different normative specifications.

3.1 Syntax of the Query Language

A query language in the case of action languages usually consists of statements describing initial conditions and statements to query the domain description w.r.t. these initial conditions. We adapt the notion of axioms for our purpose.

Definition 15 (Axiom)**.** Let $\mathcal{N}$ be a normative specification and l a ground physical literal or a ground obligation. An *axiom* is of the form **initially** l. Given a set of axioms Γ, a physical state σ in $T_{\mathcal{N}}$ *satisfies* Γ if, for every physical literal l, (**initially** l) $\in \Gamma$ implies $l \in \sigma$.

Let δ be the set of obligations d such that (**initially** d) $\in \Gamma$. A set of axioms Γ is an *initial specification for* $\mathcal{N}$ if, for every physical state σ that satisfies Γ, σ/δ forms a state of $T_{\mathcal{N}}$. Such states σ/δ are called *initial w.r.t.* Γ.

We thus specify that an initial specification for $\mathcal{N}$ aligns with Def. 5, i.e., if Γ contains an axiom for an obligation to achieve (maintain) l, then it must also contain an axiom for $\neg l$ (l). Note that a set of axioms may not *fully specify* the physical state σ, i.e., there may be several states σ that satisfy Γ, hence several initial states.

An *action sequence* is a finite sequence $((A_1,t_1), \ldots, (A_k,t_k))$ such that, for all i with $1 \leq i \leq k$, A_i is a non-empty action, and $t_1, \ldots, t_k \in \mathcal{T}$ with $0 < t_1 < \cdots < t_k$. Given an action sequence, queries are defined as follows:

Definition 16 (Query)**.** Let l be a deontic literal, a deontic event—both without any occurrence of **now**—or a physical literal, $t_\alpha, t_\beta \in \mathcal{T}$ with $0 \leq t_\alpha \leq t_\beta$, and S an action sequence. A *query* is of the form $l : [t_\alpha, t_\beta] : S$.

Note that even though our query language is quite simple, it is rather versatile and allows for expressive queries due to the usage of variables in queries. Not only may we query for non-ground fluents occurring in a certain time interval, such as whether a user had some book in her possession, but also whether there occurred any obligation or violation in a given time interval without having to specify the deadline.

3.2 Semantics of the Query Language

The semantics of the query language is defined w.r.t. paths of the transition system $T_{\mathcal{N}}$. First, we establish that a path P of the form (9) *satisfies* an initial specification Γ for $\mathcal{N}$ if σ_0/δ_0 is an initial state relative to Γ. The idea is to restrict the paths considered to answer a query to those which match the initial specification.

Next, we link the action sequence in a query to a path by matching each pair (A_i, t_i) in the sequence to exactly one in the path. All other actions in the path have to be empty, i.e., they occur due to deontic events.

Definition 17 (Satisfiability of an Action Sequence)**.** Let S be an action sequence $(A'_1, t'_1), \ldots, (A'_k, t'_k)$ and P a path of the form (9). *P satisfies S* if there is an injective mapping $\mu : \{1, \ldots, k\} \mapsto \{1, \ldots, n\}$ (from S to P) such that

1. for each i with $1 \leq i \leq k$, $A'_i = A_{\mu(i)}$ and $t'_i = t_{\mu(i)}$,
2. for each j with $1 \leq j \leq n$, if $\mu(i) \neq j$ for all i with $1 \leq i \leq k$, then $A_j = \emptyset$.

Given the definition of action sequences and paths, if such an injective mapping μ exists, then it is clearly unique, and so is the path corresponding to an action sequence for a fixed initial state.

To evaluate whether a certain literal or event holds while executing a sequence of actions, we need to collect all states that fall into the time interval $[t_\alpha, t_\beta]$ given in the query. That is, we collect the state at t_α and all the states inside the interval, or alternatively the final state in the path if the last transition occurs before t_α. In the former case, if there is no action occurring precisely at t_α, then we have to consider the state prior to t_α, because that is then the current state at t_α. Formally, given a path P of the form (9) and time points $t_\alpha \leq t_\beta$, we define the set $s(P, [t_\alpha, t_\beta]) = \{\sigma_i/\delta_i \mid t_i < t_\alpha < t_{i+1}\} \cup \{\sigma_i/\delta_i \mid t_\alpha \leq t_i \leq t_\beta\} \cup \{\sigma_n/\delta_n \mid t_n < t_\alpha\}$. Additionally, we want to ensure that only those paths are considered that cover the entire interval so that we do not miss any states. Therefore, we define that path P *reaches* time point t if either $t_n \geq t$ or $\mathsf{ltp}(\delta_n) = \infty$.

Finally, we can define how queries are evaluated.

Definition 18 (Query satisfaction)**.** Let Q be a query of the form $l : [t_\alpha, t_\beta] : S$, $\mathcal{N}$ a normative specification and Γ an initial specification for $\mathcal{N}$. *Q is a consequence of Γ w.r.t. $\mathcal{N}$*, denoted by $\Gamma \models_{\mathcal{N}} Q$, if, for every path P that satisfies Γ and S and that reaches t_β, there exists a variable assignment z such that one of these conditions holds:

(a) for some $\sigma/\delta \in s(P, [t_\alpha, t_\beta])$, $\sigma/\delta \models l|_{\mathsf{z}}$ if l is a literal;

(b) for some j with $t_\alpha \leq t_j \leq t_\beta$, $\langle \delta_{j-1}, \sigma_j \rangle \vdash_{A_j, t_j} e|_{\mathsf{z}}$ if l is a deontic event.

Note that our definition of query satisfaction implies that if the action sequence is not executable, then the query holds automatically for all paths in the transition system satisfying the conditions, simply because there are none. That is related to the question of consistent action descriptions [Zhang *et al.*, 2002] and also implicit domain constraints [Herzig and Varzinczak, 2007; Thielscher, 2011], and we refer to the literature for ways to avoid such problems.

Example 19. Recall Example 4 and $\Gamma = \{\textbf{initially } \mathit{ugrad}\}$:

$$Q_1 = \mathsf{VO}^{\mathsf{d}}_X\, \mathit{ret}(b) : [1, 8] : \langle (\mathit{borrow}(b) : 1), (\mathit{ret}(b) : 4) \rangle;$$

$$Q_2 = \mathsf{O}^{\mathsf{d}}_5\, \mathit{ret}(Y) : [0, 4] : \langle (\mathit{borrow}(b) : 1) \rangle;$$

$$Q_3 = \mathit{ugrad} : [0, 9] : \langle (\mathit{borrow}(b) : 1), (\mathit{ret}(b) : 4) \rangle.$$

We obtain that $\Gamma \not\models_{\mathcal{N}} Q_1$, but $\Gamma \models_{\mathcal{N}} Q_2$ and $\Gamma \models_{\mathcal{N}} Q_3$.

We analyze decidability and computational complexity of answering queries where we measure the input in the size of the set of axioms Γ.

Theorem 20. *Let Q be a query, $\mathcal{N}$ a normative specification and Γ an initial specification for $\mathcal{N}$. If the physical states in $T_{\mathcal{N}}$ are finite, then answering $\Gamma \models_{\mathcal{N}} Q$ is decidable in* coNP. *If Γ additionally fully specifies σ, then answering $\Gamma \models_{\mathcal{N}} Q$ is in* P.

3.3 Equivalence

Equivalence is an important problem in the area of normative systems. It can be used, for example, for simplifying normative systems, which usually tend to have redundant norms. In our approach, we define equivalence of normative specifications w.r.t. the answers they provide to queries.

Definition 21 (Equivalence)**.** We say that normative specifications $\mathcal{N}_1, \mathcal{N}_2$ are *equivalent* if for every set of axioms Γ and every query Q, $\Gamma \models_{\mathcal{N}_1} Q$ if and only if $\Gamma \models_{\mathcal{N}_2} Q$.

We can show that two normative specifications being equivalent is the same as them having the same transition system.

Theorem 22. *The following conditions are equivalent for any normative specifications $\mathcal{N}_1$, $\mathcal{N}_2$:*

1) $\mathcal{N}_1$, $\mathcal{N}_2$ are equivalent.

2) $T_{\mathcal{N}_1} = T_{\mathcal{N}_2}$.

3) The sets of paths of $T_{\mathcal{N}_1}$ and of $T_{\mathcal{N}_2}$ coincide.

A stronger notion of equivalence requires equivalence in the presence of additional norms, important when modularly analyzing subsets of norms of a larger system. Two strongly equivalent subsets of a normative specification can be safely replaced by one another.

Definition 23 (Strong equivalence)**.** We say that normative specifications $\mathcal{N}_1, \mathcal{N}_2$ are *strongly equivalent* if for every normative specification $\mathcal{N}$, $\mathcal{N}_1 \cup \mathcal{N}$ is equivalent to $\mathcal{N}_2 \cup \mathcal{N}$.

Strong equivalence implies equivalence but not vice-versa.

Theorem 24. *Let $\mathcal{N}_1$, $\mathcal{N}_2$ be normative specifications. If $\mathcal{N}_1$ is strongly equivalent to $\mathcal{N}_2$, then $\mathcal{N}_1$ is also equivalent to $\mathcal{N}_2$, but the converse implication does not hold.*

4 Conclusions

We have extended Action Language $\mathcal{A}$ with features that allow reasoning about norms, time and deadlines in dynamic domains. We have shown how our language can be used to express norms involving obligations with deadlines that explicitly refer to time and actions, including obligations to-do, to-achieve and to-maintain but also contrary-to-duty situations, which previous action languages and their extensions to norms did not cover. We have defined a semantics for this language and a query language along with its semantics. Moreover, we studied the complexity and equivalence of normative specifications.

Notably, our framework may serve as a basis for introducing norms to other AI action formalisms where norms with explicit time deadlines and contrary-to-duty obligations have received little consideration so far. Interesting examples include the Event Calculus [Kowalski and Sergot, 1986], the Situation Calculus [Reiter, 1991], the Fluent Calculus [Thielscher, 1999] and extensions of Dynamic Logic [Harel, 1979] that have a solution to the frame problem [Zhang and Foo, 2001; Zhang and Foo, 2002; Castilho *et al.*, 2002; Demolombe *et al.*, 2003].

Our query language can be used to define interesting planning problems, such as finding plans which prevent violations, or whose violations are within certain limits. Additionally, our language has important applicability in the development of electronic institutions. Electronic institutions are virtual entities that maintain, promote and enforce a set of norms. They observe agent's actions to determine norm violations (resp. satisfactions), e.g., to enforce sanctions (resp. give rewards). Given its formal semantics, and its strong links to dynamic systems, $\mathcal{A}_{\mathcal{N}}$ can be used as the language to specify and disseminate the norms and the query language used to determine violations and satisfactions.

Related work on normative systems resulted in frameworks that combine obligations and time. The proposals in [Dignum and Kuiper, 1997; Dignum and Kuiper, 1998; Broersen and Brunel, 2007; Balbiani *et al.*, 2009], which combine dynamic, deontic and temporal logic, have a rich language, but they have difficulties in dealing with the frame problem, relevant in the propagation of obligations that have not been fulfilled yet [Broersen and Brunel, 2007], and with dealing with contrary-to-duty obligations. Also, no axiomatization exists for the proposals in [Dignum and Kuiper, 1997; Dignum and Kuiper, 1998], and hence automatic reasoning is not possible, while the approaches in [Broersen and Brunel, 2007; Balbiani *et al.*, 2009] do not deal with actions. In [Ågotnes *et al.*, 2010], robustness of normative systems is studied building on temporal logic, but neither deadlines nor contrary-to-duty obligations are considered. The work in [Governatori and Rotolo, 2011] aims at studying the dynamics of normative violations. However, without an explicit representation of actions, they cannot properly deal with obligations to-do, nor integrate the normative part of the system with the dynamics resulting from the execution of actions provided by Action Languages. Finally, in [Dastani *et al.*, 2012] the focus is set on an operational semantics to be able to modify a normative system during runtime. Yet, there are no time deadlines. Instead, deadlines are state conditions, which may be an interesting extension of our work, but does not cover the expressiveness provided by our formalism.

Our work opens several interesting paths for future research. First of all, we would like to design an implementation. Of course, an encoding in ASP is always possible, but perhaps more efficient solutions exist. We would also like to extend the language with other deontic constructs such as *prohibition* and *permission*. We already have some notion of prohibition, since an obligation to-maintain $\neg l$ can be seen as a prohibition to bring about l, and some notion of permission, since the removal of an obligation to-maintain $\neg l$ can be seen as a weak permission to bring about l. On the other hand, the counterpart of obligations to-do, forbidden actions, has not been considered here. Accommodating forbidden actions would require a new normative fluent $\mathsf{F}_t\, a$ meaning that action a is forbidden until time t. Moreover, we may consider extending our framework to more expressive Action Languages, more complex deadlines, and actions with different durations.

Acknowledgments We would like to thank the anonymous reviewers whose comments helped to improve the paper.

Matthias Knorr, João Leite and Martin Slota were partially supported by Fundação para a Ciência e a Tecnologia under project "ERRO – Efficient Reasoning with Rules and Ontologies" (PTDC/EIA-CCO/121823/2010). Matthias Knorr was also partially supported by FCT Grant SFRH/BPD/86970/2012 and Ricardo Gonçalves by FCT Grant SFRH/BPD/47245/2008.

References

[Ågotnes *et al.*, 2010] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Robust normative systems and a logic of norm compliance. *Logic Journal of the IGPL*, 18(1):4–30, 2010.

[Alechina *et al.*, 2012] Natasha Alechina, Mehdi Dastani, and Brian Logan. Programming norm-aware agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, volume 2 of *AAMAS '12*, pages 1057–1064. IFAAMAS, 2012.

[Artikis *et al.*, 2009] Alexander Artikis, Marek Sergot, and Jeremy Pitt. Specifying norm-governed computational societies. *ACM Trans. Comput. Log.*, 10(1):1–42, 2009.

[Balbiani *et al.*, 2009] Philippe Balbiani, Jan Broersen, and Julien Brunel. Decision procedures for a deontic logic modeling temporal inheritance of obligations. *Electr. Notes Theor. Comput. Sci.*, 231:69–89, 2009.

[Boella and van der Torre, 2004] Guido Boella and Leendert W. N. van der Torre. Regulative and constitutive norms in normative multiagent systems. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2-5, 2004*, pages 255–266. AAAI Press, 2004.

[Boella *et al.*, 2008] Guido Boella, Leendert van der Torre, and Harko Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10, 2008.

[Broersen and Brunel, 2007] Jan Broersen and Julien Brunel. Preservation of obligations in a temporal and deontic framework. In Edmund Durfee, Makoto Yokoo, Michael Huhns, and Onn Shehory, editors, *Autonomous Agents and Multi-Agent Systems*, page 177, 2007.

[Bulling and Dastani, 2011] Nils Bulling and Mehdi Dastani. Verifying normative behaviour via normative mechanism design. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, volume 1 of *IJCAI'11*, pages 103–108. AAAI Press, 2011.

[Carmo and Jones, 2002] José Carmo and Andrew Jones. Deontic logic and contrary-to-duties. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 8, pages 265–343. Kluwer Academic Publishers, Dordrecht, Holland, 2002.

[Castilho *et al.*, 2002] Marcos A. Castilho, Andreas Herzig, and Ivan José Varzinczak. It depends on the context! a decidable logic of actions and plans based on a ternary dependence relation. In Salem Benferhat and Enrico Giunchiglia, editors, *NMR*, pages 343–348, 2002.

[Cholvy, 1999] Laurence Cholvy. Checking regulation consistency by using SOL-resolution. In *ICAIL*, pages 73–79, 1999.

[Craven and Sergot, 2008] Robert Craven and Marek Sergot. Agent strands in the action language nC+. *J. Applied Logic*, 6(2):172–191, 2008.

[Dastani *et al.*, 2012] Mehdi Dastani, John-Jules Ch. Meyer, and Nick A. M. Tinnemeier. Programming norm change. *Journal of Applied Non-Classical Logics*, 22(1-2):151–180, 2012.

[Demolombe *et al.*, 2003] Robert Demolombe, Andreas Herzig, and Ivan José Varzinczak. Regression in modal logic. *Journal of Applied Non-Classical Logics*, 13(2):165–185, 2003.

[Dignum and Kuiper, 1997] Frank Dignum and Ruurd Kuiper. Combining dynamic deontic logic and temporal logic for the specification of deadlines. In *HICSS (5)*, pages 336–346, 1997.

[Dignum and Kuiper, 1998] Frank Dignum and Ruurd Kuiper. Obligations and dense time for specifying deadlines. In *HICSS (5)*, pages 186–195, 1998.

[Esteva *et al.*, 2001] Marc Esteva, Juan A. Rodríguez-Aguilar, Carles Sierra, Pere Garcia, and Josep Lluís Arcos. On the formal specifications of electronic institutions. In Frank Dignum and Carles Sierra, editors, *Agent Mediated Electronic Commerce, The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Computer Science*, pages 126–147. Springer, 2001.

[Gelfond and Lifschitz, 1998] Michael Gelfond and Vladimir Lifschitz. Action languages. *Electron. Trans. Artif. Intell.*, 2:193–210, 1998.

[Gelfond and Lobo, 2008] Michael Gelfond and Jorge Lobo. Authorization and obligation policies in dynamic systems. In Maria de la Banda and Enrico Pontelli, editors, *ICLP*, volume 5366 of *Lecture Notes in Computer Science*, pages 22–36. Springer, 2008.

[Giunchiglia *et al.*, 2004] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153(1):49–104, 2004.

[Governatori and Rotolo, 2011] Guido Governatori and Antonino Rotolo. Justice delayed is justice denied: Logics for a temporal account of reparations and legal compliance. In João Leite, Paolo Torroni, Thomas Ågotnes, Guido Boella, and Leon van der Torre, editors, *CLIMA*, volume 6814 of *Lecture Notes in Computer Science*, pages 364–382. Springer, 2011.

[Harel, 1979] David Harel. *First-Order Dynamic Logic*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1979.

[Herzig and Varzinczak, 2007] Andreas Herzig and Ivan José Varzinczak. Metatheory of actions: Beyond consistency. *Artif. Intell.*, 171(16-17):951–984, 2007.

[Kowalski and Sergot, 1986] Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New generation computing*, 4(1):67–95, 1986.

[Reiter, 1991] Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Vladimir Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[Sadiq *et al.*, 2007] Shazia Wasim Sadiq, Guido Governatori, and Kioumars Namiri. Modeling control objectives for business process compliance. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings*, volume 4714 of *Lecture Notes in Computer Science*, pages 149–164. Springer, 2007.

[Thielscher, 1999] Michael Thielscher. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial intelligence*, 111(1):277–299, 1999.

[Thielscher, 2011] Michael Thielscher. A unifying action calculus. *Artif. Intell.*, 175(1):120–141, 2011.

[Zhang and Foo, 2001] Dongmo Zhang and Norman Y. Foo. EPDL: A logic for causal reasoning. In Bernhard Nebel, editor, *IJCAI*, pages 131–138. Morgan Kaufmann, 2001.

[Zhang and Foo, 2002] Dongmo Zhang and Norman Y. Foo. Interpolation properties of action logic: Lazy-formalization to the frame problem. In Sergio Flesca, Sergio Greco, Nicola Leone, and Giovambattista Ianni, editors, *JELIA*, volume 2424 of *Lecture Notes in Computer Science*, pages 357–368. Springer, 2002.

[Zhang *et al.*, 2002] Dongmo Zhang, Samir Chopra, and Norman Y. Foo. Consistency of action descriptions. In Mitsuru Ishizuka and Abdul Sattar, editors, *PRICAI*, volume 2417 of *Lecture Notes in Computer Science*, pages 70–79. Springer, 2002.