

Robert Trypuz
Piotr Kulicki

A NORM-GIVER MEETS DEONTIC ACTION LOGIC

Abstract. In the paper we present a formal system motivated by a specific methodology of creating norms. According to the methodology, a norm-giver before establishing a set of norms should create a picture of the agent by creating his repertoire of actions. Then, knowing what the agent can do in particular situations, the norm-giver regulates these actions by assigning deontic qualifications to each of them. The set of norms created for each situation should respect (1) generally valid deontic principles being the theses of our logic and (2) facts from the ontology of action whose relevance for the systems of norms we postulate.

Keywords: deontic action logic, lawmaking, situations, agents' abilities.

Introduction

Deontic logic can be seen as a formal tool for analysing rational agent's behaviour in the context of systems of norms.¹ The interaction can be considered locally, from the perspective of an agent or globally, from the point of view of a norm-giver. In the present paper we focus on the former.

We assume the following:

- We work within logic in which deontic notions are attached to *actions*. The paper is in this sense a continuation of the work from [7, 10, 11,

¹ We would like to thank the anonymous reviewer of Logic and Logical Philosophy for his/her very constructive and detailed comments.

[12]. The present paper extends the earlier studies by the explicit introduction of multiple situations into the considerations.

- Situations are analysed separately. The history of agent’s behaviour does not influence the deontic qualifications of present actions.
- We consider only one type of agents. It means that every agent has the same abilities and the same rights and duties.
- The norm-giver has a complete knowledge about the agent’s abilities in all possible situations. If such an assumption is unrealistic for some applications, we treat the system of norms as dependant on the norm-giver’s beliefs.
- We restrict ourselves to situations with a finite number of possible actions. If there are infinite possible behaviours we have to group them into a finite number of types.

We suggest starting to build the system of norms from analysing the situations (or types of situations) separately. For any of them the norm-giver identifies the actions that are ontologically possible. Then deontic qualifications are attached to the actions. Finally, the qualifications are generalised in order to obtain general deontic rules which build the system of norms.

There may be actions that are possible in one situation and not in another. We understand that the agent has, in general, the ability to perform an action when there is at least one situation in which the action is possible. For example, we would say that an agent has the ability to turn a light on, when at least in one situation the agent can successfully do that. However, in certain situations it may be impossible because the light is already on or the light bulb is burnt out.

In order to generalise norms over situations one has to recognise the same action in different situations. To establish such a correspondence we treat actions as combinations of simple elements, such as *taking a left turn*, *answering a phone call*, *turning the light on*. These elements occur in different situations and can be relatively easily recognised there.

Such a vision of norm development can be used for establishing new systems of norms and analysing the existing ones. Its formalisation enables us to discuss several issues connected with the relation between norms and abilities, especially the well known Kant’s Law. We can also put new light on the notions of internal and external actions discussed in philosophy and AI, e.g. [5, 4].

Firstly, we present our intuitions in terms of frames which are “mathematical pictures of ontologies” [1]. Then we define an appropriate de-

ontic action logic. Finally, we apply the logic to an example taken from the area of traffic regulations.

1. Deontic action and situation frame

A *deontic action and situation frame* is a structure

$$\mathcal{DAS} = \langle \mathcal{E}, S, exe, \mathcal{LEG}, \mathcal{ILL}, \mathcal{REQ} \rangle$$

\mathcal{E} is a nonempty set. Elements of the set represent all the possible modes in which actions can be executed. Every subset of \mathcal{E} we shall call an *action*. S is a set of situations or states, understood as complete (with reference to selected properties) descriptions of the situation.²

$exe: S \rightarrow 2^{\mathcal{E}}$ is an executability function which assigns a set of modes of actions which can occur in s to every situation $s \in S$. Thus, for every s , exe indicates whether certain action mode can or cannot take place in it. We use function exe instead of (action-labeled) accessibility relation between situations, since, as mentioned in the introduction, we are not interested here in how situations are ordered.

We say that action mode e is *external* in $s \in S$ iff $e \in exe(s)$. Otherwise it remains only *internal* in s . Thus, an internal action mode in principle belongs to a *repertoire of actions*³ but it may not be executable in a certain situation.

We can extend the notion of external action mode to actions by defining external action. We say that action $X \subseteq \mathcal{E}$ is *external in state* s iff $X \cap exe(s) \neq \emptyset$ (i.e. X contains at least one external action modes in s). Otherwise the action is internal in s .

The set of all action modes external in s can be interpreted as the *local universal action* (the universal action in s). The *universal action* is the sum of all local universal actions:

$$\mathcal{E}xe = \bigcup_s exe(s).$$

² In some contexts situations can be reduced to the subsets of $2^{\mathcal{E}}$ (i.e. to the sets of actions). This way of thinking was used by Trypuz in [9]. In the approach from the present paper, two situations “consisting of” the same modes of actions may differ due to different deontic classification of those modes or simply different facts may hold in them. Thus, the function exe is introduced in order to connect situations with the appropriate sets of actions.

³ We have borrowed the term from the works of Krister Segerberg (cf. [6, 8]).

We assume that $exe(s)$ is never empty

$$\forall s \in S, exe(s) \neq \emptyset$$

and every action mode is external in at least one situation:

$$\mathcal{E} = \mathcal{E}xe. \quad (1)$$

The last assumption excludes from our consideration these agent abilities which are not manifested in any situation from S . It is worth noting that (1) is a natural result of our methodological assumption that the norm-giver creates a picture of the agent's repertoire of action by analysing what the agent can do in each particular situation.

We shall call an action X *non-empty* iff $X \neq \emptyset$. A non-empty action has to be external in at least one situation but does not have to be external in all situations.

Now we shall turn to questions about some details of the model.⁴ The first question is whether there is a situation in which $exe(s) = \mathcal{E}$. In fact this is a specification of more general issue whether for every $X \subseteq \mathcal{E}$ there is s s.t. $X = exe(s)$. This is not required, though possible. Another interesting question goes as follow: "Since every subset X of \mathcal{E} is an action, how then to be understood action $X = exe(s_1) \cap exe(s_2)$ where $s_1 \neq s_2$?" It could be said that $exe(s_1) \cap exe(s_2)$, if not impossible (i.e. not empty), is an action which can be carried out in s_1 and s_2 .

$\mathcal{L}\mathcal{E}\mathcal{G}_s$, $\mathcal{I}\mathcal{L}\mathcal{L}_s$ and $\mathcal{R}\mathcal{E}\mathcal{Q}_s$ are subsets of $2^{exe(s)}$ and should be understood as sets of legal, illegal and required actions in s , respectively. The basic characterisation of $\mathcal{L}\mathcal{E}\mathcal{G}_s$ and $\mathcal{I}\mathcal{L}\mathcal{L}_s$, for any $s \in S$ and $X, Y \in 2^{exe(s)}$, is the fact that they are ideals, i.e. they satisfy the following two principles:

$$X \in \mathcal{L}\mathcal{E}\mathcal{G}_s (\mathcal{I}\mathcal{L}\mathcal{L}_s) \text{ and } Y \subseteq X \implies Y \in \mathcal{L}\mathcal{E}\mathcal{G}_s (\mathcal{I}\mathcal{L}\mathcal{L}_s),$$

$$X \in \mathcal{L}\mathcal{E}\mathcal{G}_s (\mathcal{I}\mathcal{L}\mathcal{L}_s) \text{ and } Y \in \mathcal{L}\mathcal{E}\mathcal{G}_s (\mathcal{I}\mathcal{L}\mathcal{L}_s) \implies X \cup Y \in \mathcal{L}\mathcal{E}\mathcal{G}_s (\mathcal{I}\mathcal{L}\mathcal{L}_s).$$

It is also assumed that $\mathcal{L}\mathcal{E}\mathcal{G}_s$ and $\mathcal{I}\mathcal{L}\mathcal{L}_s$ have nothing in common:

$$\mathcal{L}\mathcal{E}\mathcal{G}_s \cap \mathcal{I}\mathcal{L}\mathcal{L}_s = \{\emptyset\}.$$

A necessary condition for two actions to be required is that their intersection should be required too:

$$X \in \mathcal{R}\mathcal{E}\mathcal{Q}_s \text{ and } Y \in \mathcal{R}\mathcal{E}\mathcal{Q}_s \implies X \cap Y \in \mathcal{R}\mathcal{E}\mathcal{Q}_s.$$

⁴ The questions were formulated by the reviewer of the paper.

An impossible action is never required:

$$\emptyset \notin \mathcal{REQ}_s. \quad (2)$$

(2) is on the ground of our framework equivalent with so called Kant's Law which would have the following shape:

$$\text{For every } X \in 2^{exe(s)}, X \in \mathcal{REQ}_s \implies X \cap exe(s) \neq \emptyset.$$

It means that some possible outcomes of the required action in s should be external in s .

Required actions are always legal:

$$\mathcal{REQ}_s \subseteq \mathcal{LEG}_s.$$

We also require that there is no situation in which all external actions are illegal:

$$exe(s) \notin \mathcal{ILL}_s. \quad (3)$$

Situation in which (3) does not hold we named in [11] *deontic hell*. Everything that the agent can do in this situation is illegal. In our approach such a trap or a moral dilemma should always be excluded by a norm giver.

For every $X, Y \in 2^{exe(s)}$, if X is an action required in s and its intersection with action Y contains no external events, then Y is illegal in s :

$$X \in \mathcal{REQ}_s \text{ and } X \cap Y \cap exe(s) = \emptyset \implies Y \in \mathcal{ILL}_s.$$

In other words, it means that if the agent (taking into account his abilities and external conditions) is required to do X in s and, at the same time, by doing X cannot bring about action $X \cap Y$ in this situation, then Y is illegal (in order not to leave the agent any doubts what a norm-giver requires from him).

2. Deontic action logic

In this section we present a logical language and a logic, deontic action logic \mathcal{DAL} , for the deontic action and situation frame. Deontic action logic is a deontic logic where deontic operators (such as obligation, permission and prohibition) apply to the names of actions. In the papers

[10, 11] we introduced a space of deontic action logics and pointed out some applications of the systems in question.

The language of \mathcal{DAL} is defined in Backus-Naur notation in the following way:

$$\varphi ::= \top \mid \alpha \doteq \alpha \mid \mathbf{P}(\alpha) \mid \mathbf{F}(\alpha) \mid \mathbf{O}(\alpha) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi \quad (4)$$

$$\alpha ::= a_i \mid \mathbf{0} \mid \mathbf{1} \mid \bar{\alpha} \mid \alpha \sqcup \alpha \mid \alpha \sqcap \alpha \quad (5)$$

where a_i belongs to a finite set of *action generators* Act_0 , “ $\mathbf{0}$ ” is the impossible action and “ $\mathbf{1}$ ” is the universal action; “ $\alpha \sqcup \beta$ ” – α or β (a free choice between α and β); “ $\alpha \sqcap \beta$ ” – α and β (parallel execution of α and β); “ $\bar{\alpha}$ ” – not α (complement of α); “ \top ” is an arbitrary tautology, “ $\alpha \doteq \beta$ ” means that α is locally (i.e. in the actual situation) identical with β (instead of $\neg(\alpha \doteq \beta)$ we will also write $\alpha \not\equiv \beta$); “ $\mathbf{P}(\alpha)$ ” – α is (strongly) permitted; “ $\mathbf{F}(\alpha)$ ” – α is forbidden, “ $\mathbf{O}(\alpha)$ ” – α is obligatory and “ $\Box\varphi$ ” – φ is necessary (in the Leibnizian sense). Further, for fixed Act_0 , by Act we shall understand the set of formulae defined by (5). Obviously, $Act_0 \subseteq Act$. Thus, operators \doteq , \mathbf{P} , \mathbf{F} and \mathbf{O} are sentence creating and take names as arguments.

We add the definitions of all missing standard PC operators and the following ones:

$$\alpha \sqsubseteq \beta =_{\text{df}} \alpha \sqcap \beta \doteq \alpha \quad (6)$$

$$\Diamond\varphi =_{\text{df}} \neg\Box\neg\varphi \quad (7)$$

$$\mathbf{P}_{we}(\alpha) =_{\text{df}} \neg\mathbf{F}(\alpha) \quad (8)$$

$$\alpha = \beta =_{\text{df}} \Box(\alpha \doteq \beta) \quad (9)$$

$$\odot(\alpha) =_{\text{df}} \alpha \not\equiv \mathbf{0} \quad (10)$$

By “ $\alpha \sqsubseteq \beta$ ” we mean that “ α is locally included in β ”. “ \Diamond ” is a dual of “ \Box ” (7). (8) defines the concept of weak permission. In contrast to a strongly permitted action, a weakly permitted one is permitted in *some* situations, in combination with *some* other actions. (9) defines global (i.e. in every context) identity between actions. (10) defines “ $\odot(\alpha)$ ”, meaning that α is executable (in the actual situation).

Now we introduce an interpretation for actions from Act . Interpretation function $\mathcal{I}: Act \rightarrow 2^{\mathcal{E}}$ is defined as follows:

$$\mathcal{I}(a_i) \subseteq \mathcal{E}, \text{ for } a_i \in Act_0,$$

$$\mathcal{I}(\mathbf{0}) = \emptyset,$$

$$\begin{aligned}
\mathcal{I}(\mathbf{1}) &= \mathcal{E}, \\
\mathcal{I}(\alpha \sqcup \beta) &= \mathcal{I}(\alpha) \cup \mathcal{I}(\beta), \\
\mathcal{I}(\alpha \sqcap \beta) &= \mathcal{I}(\alpha) \cap \mathcal{I}(\beta), \\
\mathcal{I}(\bar{\alpha}) &= \mathcal{E} \setminus \mathcal{I}(\alpha).
\end{aligned}$$

Thus, every action from the language is interpreted as an action from \mathcal{DAS} , i.e. a set of modes of actions (α 's extension), and operations “ \sqcup ”, “ \sqcap ” between actions and “ $\bar{}$ ” on a single action are interpreted as the set-theoretical operations on actions from the frame. If $\mathcal{I}(\alpha) \neq \emptyset$, then it is a non-empty action. It is worth noting that introducing the interpretation function \mathcal{I} we assumed that we are able to give the same name to different modes of actions and actions occurring in different situations. By $\mathcal{I}_s(\alpha)$ we understand a local interpretation of action (i.e. relativised to situation s) and define it as follows:

$$\mathcal{I}_s(\alpha) = \mathcal{I}(\alpha) \cap \text{exe}(s).$$

$\mathcal{I}_s(\alpha) = \emptyset$ means that $\mathcal{I}(\alpha)$ is not an external action in s , whereas $\mathcal{I}_s(\alpha) \neq \emptyset$ states that it is. Sometimes we will say that α is an external action in s if $\mathcal{I}_s(\alpha) \neq \emptyset$. Thus $\mathcal{I}_s(\alpha)$ points out those modes of action (named) α which can be carried in s . Of course logic cannot explain why $\mathcal{I}_s(\alpha)$ consists of these and those elements from $\mathcal{I}(\alpha)$. The interpretation \mathcal{I} constitutes a global semantics that fixes the meaning of actions independently from situation and the function $\text{exe}(s)$ constitutes a local ontology for a state s . \mathcal{I}_s is a combination of them.

Here we should raise the question of the interpretations of Boolean algebra. We shall introduce **BA** on two levels, a local and a global one. The local **BA** is always relativised to a certain situation and contains the action generators which are external in it and **BA** built by means of these generators (subset of Act_o). We shall treat it as a description of all possible choices which the agent has in this situation. It means that each situation has its own **BA** described by means of local identity ($\overset{\circ}{=}$).

By generalisation we obtain the global **BA** which applies to actions seen from the bird perspective, i.e. it describes all actions and relations between them relativising in a sense all situations to one. At this level we have all the necessary logical relations between actions (e.g. “ $\square(\alpha \sqcup \bar{\alpha} \overset{\circ}{=} \mathbf{1})$ ”) and the (necessary) ontological statements (e.g. “ $\square \neg \odot (\textit{turn-right} \sqcap \textit{turn-left})$ ”). The set of action generators here is a sum of all of generators from local **BA**. The necessary logical and

ontological statements would express an ontology of norms and actions and would be used in verification of statements for every situation.

It is a well-known fact about **BA** in general that if a set of generators is finite, then there are elements of **BA** called *atoms* such that there is no element between them and **0**. In the case of **BA** of actions we get atomic actions. Atoms in local **BA**s can be different, because sets of actions possible in different situations are different.

Every atom is a combination of all action generators of **BA** and has the form:

$$\delta_1 \sqcap \dots \sqcap \delta_n,$$

where δ_k is a generator $a_k \in X(\subseteq Act_0)$ or its complement. Following [2], we accept that the interpretation of every atom is a singleton (two lines over the set denote its cardinality):

$$\overline{\overline{\mathcal{I}(\delta)}} = 1,$$

where δ is an atom of **BA**. A basic intuition is such that an atomic action corresponding to (a set with) one event/outcome is *deterministic*. As it was noted by Castro and Maibaum in [2], *indeterminism of actions* may result from combining atomic (deterministic) actions through the free choice operator (\sqcup).

A class of models consisting of the models from \mathcal{M} with deontic action and situation frame \mathcal{DAS} together with interpretation and standard valuation function will be represented by \mathbf{C} .

Satisfaction conditions for the formulae of \mathcal{DAL} in any model $\mathcal{M} \in \mathbf{C}$ are defined below:

$$\begin{array}{ll} \mathcal{M}, s \models \mathbf{P}(\alpha) & \iff \mathcal{I}_s(\alpha) \in \mathcal{LEG}_s \\ \mathcal{M}, s \models \mathbf{F}(\alpha) & \iff \mathcal{I}_s(\alpha) \in \mathcal{ILL}_s \\ \mathcal{M}, s \models \mathbf{O}(\alpha) & \iff \mathcal{I}_s(\alpha) \in \mathcal{REQ}_s \\ \mathcal{M}, s \models \top & \\ \mathcal{M}, s \models \alpha \doteq \beta & \iff \mathcal{I}_s(\alpha) = \mathcal{I}_s(\beta) \\ \mathcal{M}, s \models \neg\varphi & \iff \mathcal{M}, s \not\models \varphi \\ \mathcal{M}, s \models \varphi \wedge \psi & \iff \mathcal{M}, s \models \varphi \text{ and } \mathcal{M}, s \models \psi \\ \mathcal{M}, s \models \Box\varphi & \iff \forall s \in S, \mathcal{M}, s \models \varphi \end{array}$$

Thus, action α is permitted (forbidden, obligatory) in situation s if and only if its local interpretation in s belongs to \mathcal{LEG}_s (\mathcal{ILL}_s , \mathcal{REQ}_s). That means in practice that if α is permitted in a situation, then it is

permitted in combination with any action (cf. theses 22 and 25) in it. The same is true for forbiddance and obligation. Satisfaction conditions for defined concepts are as follows:

$$\begin{aligned}
\mathcal{M}, s \models P_{we}(\alpha) &\iff \mathcal{I}_s(\alpha) \notin \mathcal{ILL}_s \\
\mathcal{M}, s \models \alpha = \beta &\iff \forall s \in S, \mathcal{I}_s(\alpha) = \mathcal{I}_s(\beta) \text{ (or } \mathcal{I}(\alpha) = \mathcal{I}(\beta)) \\
\mathcal{M}, s \models \diamond\varphi &\iff \exists s \in S, \mathcal{M}, s \models \varphi \\
\mathcal{M}, s \models \odot(\alpha) &\iff \mathcal{I}_s(\alpha) \neq \emptyset
\end{aligned}$$

Axiomatisation. \mathcal{DAL} is axiomatised by the following set of axioms and rules: axioms of PC, S5 axioms for “ \square ”, indegenerate Boolean algebra (**BA**) of actions from Act , identity axioms:

$$\alpha \doteq \alpha \tag{11}$$

$$\alpha \doteq \beta \rightarrow (\varphi \rightarrow \varphi(\alpha/\beta)), \tag{12}$$

where φ is any sentence not containing \square or \diamond and $\varphi(\alpha/\beta)$ is a sentence obtained from φ by replacing some or all the occurrences of α with β , and, finally specific axioms for deontic operators:

$$P(\alpha \sqcup \beta) \equiv P(\alpha) \wedge P(\beta) \tag{13}$$

$$F(\alpha \sqcup \beta) \equiv F(\alpha) \wedge F(\beta) \tag{14}$$

$$\alpha \doteq \mathbf{0} \equiv F(\alpha) \wedge P(\alpha) \tag{15}$$

$$\neg F(\mathbf{1}) \tag{16}$$

$$\mathbf{0}(\alpha) \wedge \mathbf{0}(\beta) \rightarrow \mathbf{0}(\alpha \sqcap \beta) \tag{17}$$

$$\mathbf{0}(\alpha) \rightarrow P(\alpha) \tag{18}$$

$$\mathbf{0}(\alpha) \wedge \alpha \sqcap \beta \doteq \mathbf{0} \rightarrow F(\beta) \tag{19}$$

The rules of Modus Ponens and Necessitation for \square are also accepted.⁵

Axiom (13) says that free choice between two actions is permitted if and only if each of them is permitted. Axiom (14) has a similar meaning for forbidden actions. (15) expresses the fact that only the impossible action is at the same time permitted and forbidden. Intuitively, for an action to be permitted (forbidden) means “permitted (forbidden) in any

⁵ The same intuitions can be formalised using other formal tools. Other approaches towards deontic action logic include the one from [3], where the first theory with modalities was presented. We prefer to use **BA** as it constitutes a ready-made ontology of actions in a form of simple and well established theory.

possible circumstances”, i.e. “in combination with any other action” (cf. thesis 22). By (16) we reject situations (called *deontic hell*) where all actions are forbidden. We also accept that if two actions are obligatory then their parallel execution is obligatory too (17) and that obligation implies permission (18). (19) is a logical counterpart of *principle of obligation economy* and concerns restrictions which obligation “ $\mathbf{O}(\alpha)$ ” imposes on other actions which cannot be done in parallel with α . We tend to believe that those actions should be forbidden.

There is a list of a few self-explanatory theses of \mathcal{DAL} below.

$$\alpha = \beta \rightarrow \alpha \stackrel{\circ}{=} \beta \quad \odot(\alpha) \rightarrow \diamond \odot(\alpha) \quad (20)$$

$$\mathbf{P}(\beta) \wedge \alpha \sqsubseteq \beta \rightarrow \mathbf{P}(\alpha) \quad \mathbf{F}(\beta) \wedge \alpha \sqsubseteq \beta \rightarrow \mathbf{F}(\alpha) \quad (21)$$

$$\mathbf{P}(\alpha) \rightarrow \mathbf{P}(\alpha \sqcap \beta) \quad \mathbf{F}(\alpha) \rightarrow \mathbf{F}(\alpha \sqcap \beta) \quad (22)$$

$$\odot(\alpha) \equiv (\mathbf{F}(\alpha) \rightarrow \neg \mathbf{P}(\alpha)) \quad (23)$$

$$\mathbf{P}(\alpha) \wedge \mathbf{F}(\beta) \rightarrow \neg \odot(\alpha \sqcap \beta) \quad (24)$$

$$\mathbf{P}(\alpha) \wedge \odot(\alpha \sqcap \beta) \rightarrow \mathbf{P}(\alpha \sqcap \beta) \wedge \neg \mathbf{F}(\beta). \quad (25)$$

$$\mathbf{O}(\alpha) \rightarrow \odot(\alpha) \quad \neg \mathbf{O}(\mathbf{0}) \quad (26)$$

$$\mathbf{O}(\alpha) \wedge \mathbf{O}(\beta) \rightarrow \odot(\alpha \sqcap \beta) \quad (27)$$

$$\neg(\mathbf{O}(\alpha) \wedge \mathbf{O}(\bar{\alpha})) \quad (28)$$

$$\neg(\mathbf{O}(\alpha) \wedge \mathbf{F}(\alpha)) \quad (29)$$

$$\mathbf{O}(\alpha) \rightarrow \mathbf{P}_{we}(\alpha) \quad (30)$$

$$\mathbf{F}(\alpha) \wedge \odot(\alpha \sqcap \beta) \rightarrow \neg \mathbf{O}(\beta) \quad (31)$$

$$\mathbf{O}(\alpha) \wedge \neg \odot(\alpha \sqcap \beta) \rightarrow \neg \mathbf{P}(\beta) \quad (32)$$

THEOREM 1. \mathcal{DAL} is sound and complete with respect to class \mathbf{C} .

PROOF. It is easy to check that the logic is sound with respect to class \mathbf{C} of models. We prove completeness in standard way (in fact we shall only sketch the proof). First we define a canonical model for \mathcal{DAL} as below.

DEFINITION 1. Let Σ be a set of theses of \mathcal{DAL} . Then $[\alpha]_{=}$ be an equivalence class of relation $=$ (note that $\alpha = \beta =_{\text{df}} \square(\alpha \stackrel{\circ}{=} \beta)$), for $\alpha \in \text{Act}$. Then a canonical model for Σ consists of:

- $S^{\Sigma} = \{\Gamma : \Sigma \subseteq \Gamma \text{ and } \Gamma \text{ is a maximal consistent set of formulas in } \Sigma\}$
- $\mathcal{E}^{\Sigma} = \{[\delta_1 \sqcap \dots \sqcap \delta_n]_{=} : \delta_k \text{ is a generator } a_k \in \text{Act}_0 \text{ or its complement}\}$

- $exe^\Sigma: S^\Sigma \longrightarrow 2^{\mathcal{E}^\Sigma}$ is whatever function which satisfies conditions:⁶
 - (1) if $\langle \Gamma, X \rangle \in exe^\Sigma$, then $X = \{[\alpha]_ = \in \mathcal{E}^\Sigma : \alpha \not\equiv \mathbf{0} \in \Gamma\}$
 - (2) $exe^\Sigma(\Gamma) \neq \emptyset$, for any $\Gamma \in S^\Sigma$.
- $\mathcal{I}^\Sigma(\alpha) = \{[\beta]_ = \in \mathcal{E}^\Sigma : \beta \sqsubseteq \alpha \in \Sigma\}$
- $\mathcal{LEG}_\Gamma^\Sigma = \{X \in 2^{exe^\Sigma(\Gamma)} : \text{for every } [\alpha]_ = \in X, \text{P}(\alpha) \in \Gamma\}$
- $\mathcal{ILL}_\Gamma^\Sigma = \{X \in 2^{exe^\Sigma(\Gamma)} : \text{for every } [\alpha]_ = \in X, \text{F}(\alpha) \in \Gamma\}$
- $\mathcal{OBL}_\Gamma^\Sigma = \{X \in 2^{exe^\Sigma(\Gamma)} : \text{for every } [\alpha]_ = \in X, \mathbf{0}(\alpha) \in \Gamma\}$

It is easy, though time consuming, to prove that \mathbf{C} contains model $\mathcal{M}^\Sigma = \langle \mathcal{DAF}^\Sigma, \mathcal{I}^\Sigma \rangle$, where $\mathcal{DAF}^\Sigma = \langle \mathcal{E}^\Sigma, \mathcal{LEG}^\Sigma, \mathcal{ILL}^\Sigma, \mathcal{OBL}^\Sigma \rangle$. Then it is enough to show that in a canonical model (defined above) the situations verify exactly the formulas they contain, i.e. that

$$\mathcal{M}^\Sigma, \Gamma \models \varphi \iff \varphi \in \Gamma,$$

for every Γ in \mathcal{M}^Σ . This lemma is called “truth lemma” and it is proved by induction on the complexity of φ . From this lemma follows that the formulas true in the canonical model are precisely the theorems of \mathcal{DAL} and that the theorems of \mathcal{DAL} are just those formulas true in any such canonical model. \square

3. Example

In order to illustrate the notions of our logic we shall answer the question: which norms are in force in the situation depicted in Figure 1:

- (A) $\text{P}(\textit{turn-right})$
- (B) $\text{F}(\textit{go-straight})$
- (C) $\mathbf{0}(\textit{turn-right} \sqcup \textit{turn-left})$

There are two basic assumptions: (1) that *any inconsistent set of norms cannot be accepted*. In practice it means that set of norms Γ might be acceptable if $\Gamma \not\vdash \perp$ and that (2) we have a “minimal” ontology of action enabling us to decide which actions (by nature) and in a particular situation can be carried out in parallel.

We see from the traffic sign that $\mathbf{0}(\textit{turn-right} \sqcup \textit{turn-left})$. Thus (C) is a correct answer. From the ontological analysis we know that in every situation any two actions can be carried out in parallel, formally:

⁶ We shall write $\varphi \in \Sigma$ meaning the same as $\vdash_\Sigma \varphi$.

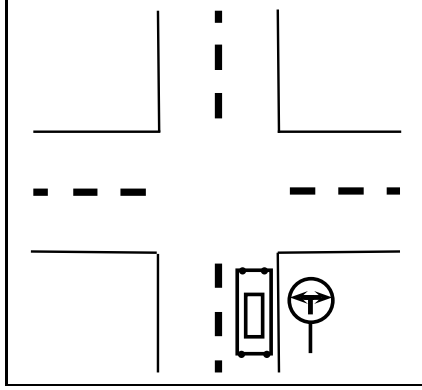


Figure 1. Crossroad 1

$\Box\neg\odot(\textit{turn-right} \sqcap \textit{turn-left})$, $\Box\neg\odot(\textit{turn-right} \sqcap \textit{go-straight})$ and $\Box\neg\odot(\textit{turn-left} \sqcap \textit{go-straight})$. (A) can be accepted, which follows from the fact that (use (13) and (18)):

$$\{\mathbf{O}(\textit{turn-right} \sqcup \textit{turn-left})\} \vdash \mathbf{P}(\textit{turn-right})$$

(B) can be accepted too because (use (19)):

$$\{\mathbf{O}(\textit{turn-right} \sqcup \textit{turn-left}), \Box\neg\odot((\textit{turn-right} \sqcup \textit{turn-left}) \sqcap \textit{go-straight})\} \\ \vdash \mathbf{F}(\textit{go-straight})$$

Now let us modify the situation s . It is much like the previous one but now the agent cannot turn left because there has been an accident and the road to the left is closed (cf. Figure 2).

Turning left belongs to the agent's repertoire, however, in the situation s it cannot be executed: $s \models \neg\odot(\textit{turn-left})$. Our logic immediately provides that $\textit{turn-left}$ cannot be obligatory since it is impossible and that $\textit{turn-right}$ is obligatory:

$$\{\mathbf{O}(\textit{turn-right} \sqcup \textit{turn-left}), \neg\odot(\textit{turn-left})\} \vdash \mathbf{O}(\textit{turn-right})$$

Thus, the facts about the situation change the norms in such a way that the agent does not have the choice between turning left or right any more.

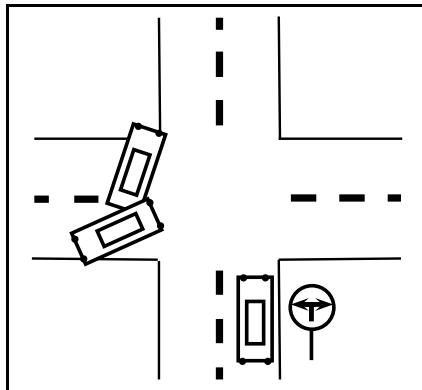


Figure 2. Crossroad 2

Conclusions

In the paper we have presented a formal system motivated by a specific methodology of creating norms. We believe that a norm-giver before establishing a set of norms should create a picture of the agent by creating his repertoire of actions. Then, knowing what the agent can do in particular situations, the norm-giver regulates these actions by assigning deontic qualifications to each of them. The set of norms created for each situation should respect generally valid deontic principles being the axioms/theses of our logic.

As the case study has shown our system can be also used for validation of existing systems of norms.

In the future work we shall explore systems that cross the limits defined by the assumptions listed in the introduction. We are especially interested in systems with different types of agents, systems in which the sequence of actions may influence their deontic qualification.

References

- [1] Blackburn, P., M. de Rijke and Y. Venema, *Modal Logic*, Cambridge University Press, 2001.
- [2] Castro, P.F., and T.S.E. Maibaum, “Deontic action logic, atomic Boolean algebras and fault-tolerance”, *Journal of Applied Logic* 7, 4 (2009): 441–466.

- [3] Gert-Jan Lokhorst, C., “Reasoning about actions and obligations in first-order logic”, *Studia Logica* 57, 1 (1996): 221–237.
- [4] Lorini, E., and Herzig, A., “A logic of intention and attempt”, *Synthese* 163, 1 (2008): 45–77.
- [5] O’Shaughnessy, B., “Trying (as the mental ‘pineal gland’)”, *Journal of Philosophy* 70 (1973): 365–386.
- [6] Segerberg, K., “Applying modal logic”, *Studia Logica*, 39, 2–3 (1980).
- [7] Segerberg, K., “A deontic logic of action”, *Studia Logica*, 41, 2–3 (1982).
- [8] Segerberg, K., “Towards an exact philosophy of actio”, *Topoi* 3, 1 (1984).
- [9] Trypuz, R., “Setna – prosta teoria norm i działań”, *Filozofia Nauki* 3–4 (2008): 155–175.
- [10] Trypuz, R., and P. Kulicki, “A systematics of deontic action logics based on Boolean algebra”, *Logic and Logical Philosophy*, 18, 3–4 (2009): 253–270.
- [11] Trypuz, R., and P. Kulicki, “Towards metalogical systematisation of deontic action logics based on Boolean algebra”, pages 132–147 in: G. Governatori and G. Sartor (eds.), *Deontic Logic in Computer Science (DEON 2010)*, 2010.
- [12] Trypuz, R., and P. Kulicki, “On deontic action logics based on Boolean algebra”, *Journal of Applied Logic* (submitted).

ROBERT TRYPUZ
Department of Logic
Faculty of Philosophy
The John Paul II Catholic University of Lublin
Al. Raclawickie 14
Lublin, Poland
trypuz@kul.pl

PIOTR KULICKI
Department of Foundation of Computer Science
Faculty of Philosophy
The John Paul II Catholic University of Lublin
Al. Raclawickie 14
Lublin, Poland
kulicki@kul.pl