Piotr Kulicki
Robert Trypuz

Connecting Actions and States in Deontic Logic

Abstract. This paper tackles the problem of inference in normative systems where norms concerning actions and states of affairs appear together. A deontic logic of actions and states is proposed as a solution. It is made up of two independent deontic logics, namely a deontic logic of action and a deontic logic of states, interlinked by bridging definitions. It is shown at a language and a model level how an agent should look for norms to follow in a concrete situation. It is pointed out that such specific norms are obtained by finding the most specific obligation and the most general prohibition. They are to be derived from all norms applicable to the situation by using the principles of the logic presented in this paper.

Keywords: Deontic action logic, Norms on actions and states, General and specific norms.

1. Introduction

'Quidquid agis, prudenter agas et respice finem'[1]—this famous quote from Ovidius points out two aspects of actions important from the deontic point of view—their course and their result.[2] Before we concentrate on them, let us focus on the distinction between general and specific norms. General norms appear in the sources of norms, such as legal documents, agreements, orders, informal social regulations, etc. They are external in relation to agents and are usually formed in an abstract way. Specific norms are connected with a particular situation of an agent. They are the result of applying (by an agent himself or by a judge) all general norms the agent should comply with in a certain situation. The distinction is present in the theory of law [10] and has recently been discussed in the context of deontic logic in [3], where the notion of obligation is used instead of the notion of specific norm.

[1] Whatever you do, do it cautiously, and with the end in mind.

[2] This paper is a revised and expanded version of a paper entitled 'A Deontic Logic of Actions and States' published in the Proceedings of DEON 2014 [17]. More precisely, the essential part of the formal theory is preserved but some formal drawbacks of the conference paper are eliminated, the intuitive description of the formal theory is amended, examples are added for clarity, more explicit justification for some claims is presented and less important issues are skipped.

Presented by **Jacek Malinowski**; *Received* June 14, 2016

Studia Logica (2017) 105: 915–942
DOI: 10.1007/s11225-017-9716-1

Among general norms there are those that concern obligated, recommended, permitted or prohibited actions (a-norms) and those that concern desired, preferred, accepted or forbidden states (s-norms) (see e.g. [2, 21]).

> In every case a norm of conduct is a pronouncement which points out for the addressee a more or less generally defined conduct in any, or under specified, circumstances. Hence, it sets him the duty of undertaking a specified action, and refraining from any other discordant with it. It also sometimes happens that a norm points out for an addressee a duty of bringing about some state of affairs without any indications of the manner in which this state of affairs is to be attained [21, chapter viii].

Both kinds of norms can be present in the same normative systems. For example, in [2] the authors point out both kinds of norms in the Spanish constitution. Further examples of the coexistence of action and state norms can also be found in everyday situations. Let us consider an agent signing a task contract to carry out construction works. In the task contract there is a short description of the desired product, i.e., a state of affairs which is to be attained. At the same time the work activities are regulated, among others, by safety standards which limit all possible actions that lead to the desired effects to those which are safe. For instance during the construction works: it is obligatory to use designated passages when moving from one place to another (never take hazardous shortcuts!), it is forbidden to throw objects, etc. In deontic logic the two types of norms are not usually present together within formal systems. Often they are regarded as linguistic variants of the same normative reality. We are interested in a deontic logic in which we can express norms of these two kinds. There are some works that tackle this problem such as [6, 19] and recently [7], but we are not fully pleased with those solutions mostly because of the fact that they do not really separate the deontic properties of actions from the properties of states. An approach closer to ours is presented in [4, 13–15], where a two-sorted propositional language is used to deal separately with the properties of states and the properties of actions. However, the deontic properties of actions and the properties of states are connected only very loosely in those works.

By actions we mean action types (not tokens)—see [20]. From the linguistic point of view we consider general names referring to actions. Names of actions are arguments of the operators of obligation and prohibition. Together with the deontic operators they make up *a-norms*. Similarly we deal with states. We do not refer directly to particular states, but we use propositions to describe them. Each proposition can be then connected to all the states in which it is true. Propositions are arguments of the operators of obligation and prohibition. Together with the deontic operators they make up *s-norm*s.

We are interested in two kinds of reasoning about norms. One is a derivation of new general norms from the general norms already accepted. Derived norms are usually obtained by a combination of norms existing within the system. We want to be able to combine two *a-norm*s together and two *s-norm*s together, but also *a-norm*s with *s-norm*s.

The other kind of normative reasoning we are interested in is discovering specific norms for a particular agent and situation in a normative environment. Ignoring a sophisticated ontological distinction between general and specific norms we attempt to find the most specific norm, in the case of obligation and the most general norm, in the case of prohibition; they are to be derived from all the norms applicable to the situation. Specific norms understood in such a way are formally of the same type as general norms. That allows us to discuss both kinds of norms in one formal system. We are interested in the possibility of expressing *a-norm*s and *s-norm*s in one framework acknowledging the fact that they are ontologically different. A separate question is whether the logical laws governing reasoning about them are the same or different. In our opinion even if the former is the case, to understand their mutual relation they should be separated.

Most of our effort is directed towards building a model in which all kinds of just mentioned norms can be defined. Two logics: deontic logic of states and deontic logic of actions corresponding to the elements of the model are then introduced. Finally on the basis of this logics operators connecting the two logics are defined. It shows up that the defined operators are adequate with respect to the constructions connecting *a-norm*s and *s-norm*s in the model (see Observation 2).

In Section 2 we introduce a model and our notion of norm within that model. In Section 3 we define a language and its interpretation in the model, and in Section 4 we introduce a logic.

2. Frames for Deontic Actions and States and a Bridge Between Them

2.1. Deontic Frames for Actions and States: $\mathcal{DAF}$ and $\mathcal{DSF}$

2.1.1. Deontic Action Frame $\mathcal{DAF}$. A deontic action frame with sets of legal and illegal actions was described in Segerberg's [12]. His results have been systematized in [16,18] and his deontic action frame has been extended by a set of required actions (corresponding to obligation operator in the language). The deontic action frame is a structure: $\mathcal{DAF} = \langle \mathcal{AF}, \mathcal{ILL}^a, \mathcal{REQ}^a \rangle$, where $\mathcal{AF}$ is an action frame being a triple[3]: $\mathcal{AF} = \langle \mathcal{W}, \mathcal{E}, \mathcal{S}tep \rangle$. $\mathcal{W}$ is a nonempty, finite set of states. States are characterized by propositions, that are true in them. We assume that there are no two different states with the same set of propositions true in both of them. In other words we can always separate worlds with the use of propositions. $\mathcal{E}$ is a nonempty, finite set of *atomic action types* used for labelling transitions between states. The same label can be used for labelling different transitions so the labels can seen as a cross-situation identification of actions. $\mathcal{S}tep$ is a nonempty finite set of transitions which we also call *action steps*. Every element of $\mathcal{S}tep$ is a triple $\langle w_1, w_2, e \rangle$, where $w_1, w_2 \in \mathcal{W}$ are initial and final states respectively and $e \in \mathcal{E}$ is a label of an action which causes the transition from w_1 to w_2. The subsets of $\mathcal{S}tep$ represent arbitrary *action types*.

Thus each action type consists of transitions that are additionally described by the labels denoting different modes of acting by which these transitions are accomplished. It is also worth stressing that one and the same set of transitions (if abstracted from the labels) can make up two different action types. For instance, one may step down a ladder or jump from its top, reaching exactly the same end state (while starting from the same initial step). The transition itself is then exactly the same, but still one may say that only the first action is permitted (see also Example 1 further in this section).

We can model various operations on action types by set theoretical operations. Joint (parallel) realization of two action types can be captured by intersection, choice—by sum and refraining from action type—by its complement. Note that the labelled system allows us to name only those action types that can be constructed as a sum of atomic action types from $\mathcal{E}$.

We do not impose any restrictions on the frame $\mathcal{AF}$. Thus, it may happen that on the one hand, we have an indeterministic execution of an action, e.g. $\langle w_1, w_2, e \rangle \in \mathcal{S}tep$ and $\langle w_1, w_3, e \rangle \in \mathcal{S}tep$ ($w_2 \neq w_3$), and, on the other hand, that the same transition is a result of the execution of two different

[3]Indeed, an action frame can be seen as a labelled transition system as described in [14]. In the action model we do not really take advantage of the information about the final state of the transitions. We work only on the information about labels as actions that can take place in a particular state. This point of view was presented and studied in [12,16]. One can also understand the action frame presented in this section as a one-state action frame. We shall, however, need the full structure introduced here to combine an action frame with state frame in order to obtain the final model.

actions, e.g. $\langle w_1, w_2, e_1\rangle \in \mathcal{S}tep$ and $\langle w_1, w_2, e_2\rangle \in \mathcal{S}tep$ ($e_1 \neq e_2$). However, transitions with the same start, end and label are identical.

$\mathcal{ILL}^a$ and $\mathcal{REQ}^a$ are defined as functions from $\mathcal{W}$ to $2^{2^{\mathcal{S}tep}}$, so sets of action types, each represented as a set of steps, are their values. $\mathcal{ILL}^a(w)$ is a set of illegal (forbidden) actions in w, whereas $\mathcal{REQ}^a(w)$—a set of required (obligatory) ones in w. We assume that each element of $\mathcal{ILL}^a(w)$ and $\mathcal{REQ}^a(w)$ encodes an *a-norm* which comes from a legal document, social practice, etc. or is inferred from other norms. Note that each situation has its own set of norms.

We use sets of sets of transitions instead of sets of transitions in a way analogous to the neighborhood semantics of modal logic. That is because of the properties of obligation which we could not describe in a simpler model.

Because of the indeterministic character of our system, we formulate the following constraint: whenever $\langle w, w_1, e\rangle$ belongs to some A in $\mathcal{ILL}^a(w)$ or $\mathcal{REQ}^a(w)$, then for any w_2, $\langle w, w_2, e\rangle$ also belongs to A. We will call it a 'deontic action consistency' constraint. It says that on the level of deontic description of actions we do not consider their results. Actions are prohibited or obligatory as actions, no matter what their results are.[4]

We believe, following natural language and legal practice [21], that the approach to prohibition and obligation should be different. When we prohibit an action type we prohibit the execution of every action token denoted by the general action name and when we prohibit bringing about a state described by a proposition we prohibit all its concrete realizations. On the one hand, sub-action or sub-proposition (viz. an action or proposition referring to a subset of action tokens or states) of a prohibited action or proposition is also forbidden. On the other hand, the obligation concerning an action name or proposition is fulfilled if any action token or state fulfilling the specification is realized. However, obligations should not be overgeneralized, i.e., the fact that a set of action tokens or states is obligatory does not entail that its supersets are also obligatory.[5] In our opinion the overgeneralized obligations are not only less 'useful' than the original ones but they are also sometimes wrong. More detailed description and justification of those intuitions can be found in [18]. They can be expressed by imposing

[4] A similar constraint is present in [13,14] under the name of the absence of 'moral luck'. We are not quite sure whether the constraint formalizes what philosophers call *moral luck* so we prefer not to use the name.

[5] For example the Ross paradox is a formula which overgeneralizes obligation and causes a loss of information. That is the reason why we intend to avoid it in our system.

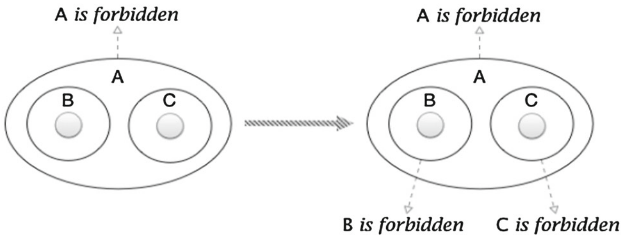

Figure 1. Any way of performing a forbidden action A is forbidden. B and C are more specific then A, so we infer that they are forbidden too

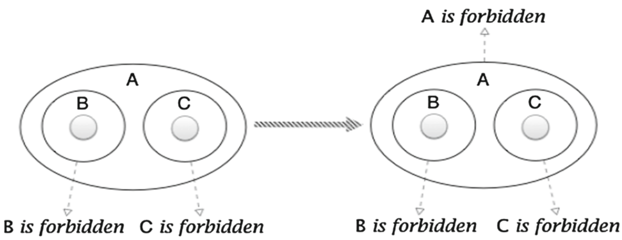

Figure 2. Prohibited actions can be aggregated. B and C being forbidden make up their sum (the action described as a choice between them) A, so we infer that A is forbidden too

ideal conditions on the set $\mathcal{ILL}^a(w)$ and the principles of agglomeration, economy and trimming which we define below.

For any $w \in \mathcal{W}$ and $A, B \in 2^{\mathcal{S}tep}$, $\mathcal{ILL}^a(w)$ satisfies the three conditions bellow making up $\mathcal{ILL}^a(w)$ to be an ideal (in the algebraic sense)—see Figures 1 and 2 for constraints (1) and (2) respectively. An impossible action, by its nature, cannot be carried out. Following Segerberg (see [12]) we assume it is forbidden.[6]

$$A \in \mathcal{ILL}^a(w) \;\&\; B \subseteq A \Longrightarrow B \in \mathcal{ILL}^a(w) \tag{1}$$

$$A \in \mathcal{ILL}^a(w) \;\&\; B \in \mathcal{ILL}^a(w) \Longrightarrow A \cup B \in \mathcal{ILL}^a(w) \tag{2}$$

$$\emptyset \in \mathcal{ILL}^a(w) \tag{3}$$

Taking into account the introduced properties of $\mathcal{ILL}^a$, in the case of forbidden actions we could talk about a set of illegal action steps instead of sets of sets. (This is not so in the case of required actions and the set $\mathcal{REQ}^a$). Loosely speaking we can say that an action step that belongs to

[6]Symbol '$\Longrightarrow$' is a metalanguage implication.

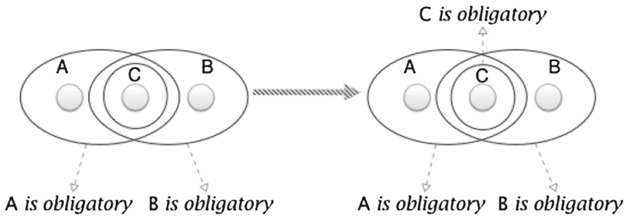

Figure 3. An intersection of two obligatory actions is also obligatory. It is because one has to comply with all their duties

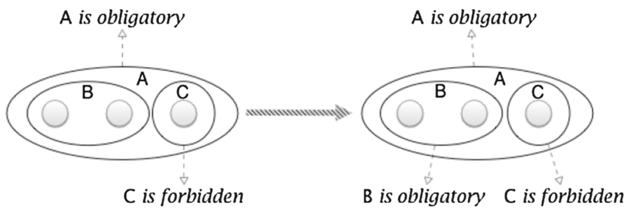

Figure 4. The forbidden ways C of realizing the obligatory action A should be excluded during practical reasoning. As a result an agent obtains a more specific (than A) action B which is obligatory and is free from illegal executions

an illegal action is illegal itself. Moreover, the set of all illegal steps is the 'largest' illegal action.

Now we move on to the characterization of obligation. We start with the so-called *agglomeration principle* (see Figure 3):

$$A \in \mathcal{REQ}^a(w) \ \&\ B \in \mathcal{REQ}^a(w) \Longrightarrow A \cap B \in \mathcal{REQ}^a(w) \tag{4}$$

We also accept the following principles, that we called in [18] *trimming* and *economy*, respectively[7]:

$$A \in \mathcal{REQ}^a(w) \ \textit{and}\ B \in \mathcal{ILL}^a(w) \Longrightarrow A \cap -B \in \mathcal{REQ}^a(w) \tag{5}$$

$$A \in \mathcal{REQ}^a(w) \Longrightarrow -A \in \mathcal{ILL}^a(w) \tag{6}$$

Figures 4 and 5 and their captions explain the meaning of the two aforementioned principles. Trimming and agglomeration principles express an idea of finding the most specific obligation that complies with all norms of the system.

It is worth noticing that agglomeration (4) follows from trimming (5) and economy (6) (see [18]).

[7] $-A$ in the formulas stands for $\mathcal{S}tep \backslash A$.

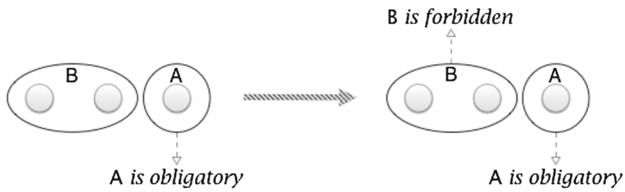

Figure 5. Whatever cannot be carried out with an obligatory action should be forbidden

Let us stress here that obligation does not ‘go up’: for $A, B \subseteq \mathcal{S}tep$, such that $A \subset B$, it is possible that $A \in \mathcal{REQ}^a(w)$ and $B \notin \mathcal{REQ}^a(w)$. Moreover, for three actions such that $A \subset B \subset C$ we can have: $A \in \mathcal{REQ}^a(w)$, $B \notin \mathcal{REQ}^a(w)$ and $C \in \mathcal{REQ}^a(w)$.

The above restrictions on the model define a logic that is in a sense minimal. Certainly more constraints can be added. Specifically we do not postulate deontic consistency in any form, thus the following properties:

$$\emptyset \notin \mathcal{REQ}^a(w) \tag{7}$$

$$A \in \mathcal{REQ}^a(w) \rightarrow A \notin \mathcal{ILL}^a(w) \tag{8}$$

are absent. That is because we want to stay open for the merge of inconsistent sets of norms.

Another possible extension would be to say that doing something is obligatory[8]:

$$\mathcal{S}tep(w) \in \mathcal{REQ}^a(w). \tag{9}$$

Moreover, we could get closer to the standard deontic logic by accepting the principle of generalization of obligation:

$$A \in \mathcal{REQ}^a(w) \ \&\ A \subseteq B \Longrightarrow B \in \mathcal{REQ}^a(w) \tag{10}$$

resulting in the presence of the Ross formula among theses.

2.1.2. Deontic State Frame $\mathcal{DSF}$. A deontic state frame is a structure

$$\mathcal{DSF} = \langle \mathcal{W}, \mathcal{REQ}_s, \mathcal{ILL}_s \rangle$$

where $\mathcal{W}$ is a set of states (as in $\mathcal{AF}$ above), $\mathcal{REQ}_s$ and $\mathcal{ILL}_s$ are functions: $\mathcal{W} \longrightarrow 2^{2^{\mathcal{W}}}$. $\mathcal{REQ}_s(w)$ and $\mathcal{ILL}_s(w)$ are sets of required and illegal sets of states (propositions) in w, respectively.

[8]By $\mathcal{S}tep(w)$ we mean the set of all transitions that start in w.

Again, as in the case of the deontic action frame, each state has its own deontic description. Thus sets of states (state propositions) are not obligatory or forbidden by themselves but from the point of view of another (current or possible) state.[9] They represent *s-norm*s and are counterparts of $\mathcal{REQ}^a(w)$ and $\mathcal{ILL}^a(w)$, satisfying the same formal conditions (Figures 1, 2, 3, 4, 5 can be applied to *s-norm*s as well as to *a-norm*s). For the sake of simplicity we adopt the same constraints for the deontic state frames as for the deontic action frames. We do it because we are mainly interested in the way the two types of norms combine in the model and in the corresponding logic. The reason for considering the two kinds of norms separately is not that they are necessarily ruled by different principles but that they are ontologically different.[10] We believe that even if the logics we combine are isomorphic the operation of combining them is by itself interesting.

However, the properties of the frames could be different. Exploring the differences would require a separate study, here we just point out a few possibilities. In the deontic action language it makes sense to accept formula (9), i.e., to state that $\mathcal{S}tep(w) \in \mathcal{REQ}^a(w)$—some action is obligatory (an agent is obliged to do something). This formula can reasonably characterize some situation w. But it has been questioned at least since von Wright's *Standard Deontic Logic* that tautology is obligatory, i.e., that $\mathcal{W} \in \mathcal{REQ}_s(w)$. Moreover, we may change the local deontic description of actions or of states into a global one which would further differentiate the approach towards the two kinds of norms. Yet another difference may occur when we introduce the operator of sequential composition of actions. Then the two fragments of our logic would have to deal with different sets of operators.

2.2. Deontic Action and State Frame $\mathcal{DASF}$

A deontic action and state frame is a structure combining the two above frames: $\mathcal{DAF}$ and $\mathcal{DSF}$:$\mathcal{DASF} = \langle \mathcal{AF}, \mathcal{ILL}^a, \mathcal{REQ}^a, \mathcal{ILL}_s, \mathcal{REQ}_s \rangle$.

Having two deontic sets concerning actions ($\mathcal{ILL}^a, \mathcal{REQ}^a$) and two sets concerning states ($\mathcal{ILL}_s, \mathcal{REQ}_s$), we intend to link them together to find all the possible combinations of actions and states regulated by norms.

On the basis of the connections between *a-norms* and *s-norms* we shall provide later new definitions of required and illegal actions taking into account the norms on states. Before that let us discuss some example.

---

[9]The connection to the neighborhood semantics is clearer here than in the case of a deontic action frame.

[10]A similar approach is accepted in Sergot's nC+ framework [4,13–15].

Example 1. *A contractor signed a contract to reconstruct a flat. The desired state of the flat is specified in the contract. The contract is a source of obligation for the contractor. The obligation concerns bringing about the state of affairs specified in the contract. Other sources of norms for the constructor are the safety regulations for construction work. For the sake of simplicity let us present it at a very general level as a prohibition to perform hazardous (unsafe) actions.*

The model of possible scenarios is depicted in the figure. There are four states: three states (w_2, w_3 and w_4) can be reached from the initial state w_1 when the deadline for the contracted work comes. The states are characterized by the following atomic propositions: deadline_came, work_done, accident_happened. In the initial state none of the atomic propositions are true. In all other states deadline_came is true. w_2 is a state in which the work is not done and an accident has not happened. w_3 is a state in which the work is done and accident has not happened. w_4 is a state in which the work is not done and an accident has happened. Atomic action types used to label the transitions are safe_action and unsafe_action. We assume (optimistically) that avoiding unsafe actions protects from accidents so there is no action step labelled safe_action leading to state w_4, in which accident_happened is true. We also assume, for the sake of simplicity, that when an accident happens the work cannot be finalized, so there is no state in which work_done and accident_happened are both true. A normative description of the situation is defined by two norms, one a-norm: it is forbidden to perform unsafe actions (unsafe_action in the figure) and one s-norm: it is obligatory to achieve the contracted state of affairs—set of all states in which work_done is true.

The means to bring about states from X starting from w is a set of action steps beginning in w and resulting in any w' in X. Formally:

$$means(w, X) \triangleq \{s \in \mathcal{S}tep{:}\exists e \in \mathcal{E}, w' \in X \text{ s.t. } s = \langle w, w', e\rangle\} \tag{11}$$

For example in Figure 6 $means(w_1, \neg work\_done)$ would be an action type consisting of the following action steps[11]: $\langle w_1, w_2, \text{safe\_action}\rangle$, $\langle w_1, w_2, \text{unsafe\_action}\rangle$ and $\langle w_1, w_4, \text{unsafe\_action}\rangle$.

It is also worth noting that $means(w, X)$ in w is always the most general action type which can be used to reach states from X. From definition (11) it also follows that the operation $means(w, X)$ is normal in the sense that

[11] For the simplicity reason we use a proposition (e.g. $\neg work\_done$) to represent a set of states; of course we mean the set of states where the proposition is true.

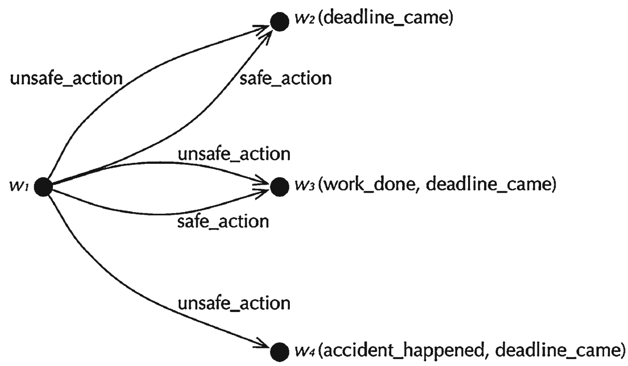

Figure 6. An example of a contract to reconstruct a flat

it distributes over operators inside it[12]:

$$means(w, W \backslash X) = Step \backslash means(w, X) \tag{12}$$

$$means(w, X \cup Y) = means(w, X) \cup means(w, Y) \tag{13}$$

$$means(w, X \cap Y) = means(w, X) \cap means(w, Y) \tag{14}$$

The model may be constructed in such a way that sets of states are not compatible with sets of actions in the sense that for some sets of state X there may be no action type $e \in \mathcal{E}$ such that $means(w, X)$ equals to the set of all $\langle w, w', e \rangle$, such that $w' \in X$. That is why combining the two models indeed expands their expressive power. Just mentioned $means(w_1, \neg work\_done)$ can be taken as an example.

A set $\mathcal{REQ}^a_s$ of required actions in the context of *s-norms* is defined as follows:

Definition 1.

$Z \in \mathcal{REQ}^a_s(w) \triangleq$

$$\text{(i) } \exists A \in \mathcal{REQ}^a(w),\ \exists X \in \mathcal{ILL}_s(w)\ (Z = A \cap means(w, -X)) \text{ or} \tag{15}$$

$$\text{(ii) } \exists A \in \mathcal{ILL}^a(w),\ \exists X \in \mathcal{REQ}_s(w)\ (Z = -A \cap means(w, X))$$

The former condition in Definition 1 states that action Z is required in w if there exists a pair (a-required action A, s-illegal state of affairs X) such that Z is a subset of A containing steps that do not lead to s-illegal states from X. The latter condition states that Z consists of means to achieve s-required states from X which does not belong to a-illegal action type A.

[12]Henceforth, we shall also use a simpler version of this formula of the following form: $means(w, -X) = -means(w, X)$.

Both conditions express the same intuition as the *trimming* principle defined separately for *a-norms* and *s-norms*. Now, however, the premisses are mixed: one of them is an *a-norm* and the other—an *s-norm*.

Let us come back to Figure 6 from the example. In Example 1 the set of states in which work_done is true is required. That set is a singleton $\{w_3\}$, thus we have that $\{w_3\} \in \mathcal{REQ}_s$. Likewise unsafe actions are forbidden, thus the set of all action steps labelled unsafe_actions is an element of $\mathcal{ILL}^a$. These two facts allow us to generate the following set of action steps that belongs to $\mathcal{REQ}^a_s$:

$$\begin{aligned}&\mathit{means}(w_1, \text{work\_done})\backslash\{\langle w_1, w, \text{unsafe\_action}\rangle{:}w \in W\}\\ &\quad= \{\langle w_1, w_3, \text{safe\_action}\rangle\}\end{aligned}$$

Observation 1. *From either of the conditions* (i) *or* (ii) *from Definition* 1 (*in the presence of economy principle for a-norms and s-norms*) *it follows that* $Z \in \mathcal{REQ}^a_s(w)$ *if:*

$$\text{(iii) } \exists A \in \mathcal{REQ}^a(w),\ \exists X \in \mathcal{REQ}_s(w)\ (Z = A \cap \mathit{means}(w, X)).$$

So Z *is required in* w *if there is a pair* (*a-required action* A*, s-required state of affairs* X) *such that* Z *is a set of action steps from* A *that are a means to bring about* X (*starting from* w).

Condition (iii) is a way to express the agglomeration principle with an *a-norm* and an *s-norm* as premises. As in the case of agglomeration within the system of *a-norms* (and *s-norms*) agglomeration can be derived from trimming and economy. Thus if economy principle were not accepted, then condition (iii) would have been added to Definition 1.

The conditions in Definition 1 do not exclude the fact that some actions in $\mathcal{REQ}^a_s(w)$ lead to illegal states in $\mathcal{ILL}_s$ and that some required states in $\mathcal{REQ}_s(w)$ are achieved by illegal actions in $\mathcal{REQ}^a(w)$. Why is it still so? Because there may be other norms applicable to the state besides those expressed by the conditions. This does not mean that the system is inconsistent. It just may be the place for applying trimming principle once more.

Loosely speaking the set $\mathcal{ILL}^a_s$ consists of actions which are a-illegal or are means to bring about s-illegal states. Formally, action types (and not action steps) are illegal here, so the definition takes the following form:

Definition 2.

$$\begin{aligned}\mathcal{ILL}^a_s(w) \triangleq \{Z \subseteq \mathit{Step}{:}&\exists A \in \mathcal{ILL}^a(w),\ \exists X \in \mathcal{ILL}_s(w),\\ &(Z = A \cup \mathit{means}(w, X))\}\end{aligned} \tag{16}$$

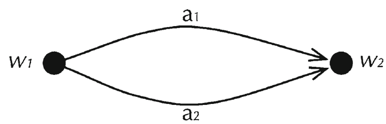

Figure 7. The only norm for the initial state w_1 be a state norm prohibiting state w_2 ($\emptyset$ is forbidden in each model). Thus $\mathcal{REQ}^a(w_1) = \mathcal{REQ}_s(w_1) = \emptyset$, $\mathcal{ILL}^a(w_1) = \{\emptyset\}$, and $\mathcal{ILL}_s(w_1) = \{\{w_2\}, \emptyset\}$

$\mathcal{ILL}^a_s$ and $\mathcal{REQ}^a_s$ satisfy most conditions analogous to the one enforced on $\mathcal{ILL}^a$ and $\mathcal{REQ}^a$, i.e., formulas (2)–(6).[13] The only exception is the counterpart of condition (1) which counterpart takes the following form:

$$A \in \mathcal{ILL}^a_s(w) \;\&\; B \subseteq A \Longrightarrow B \in \mathcal{ILL}^a_s(w) \tag{17}$$

By Definition 2 A is a set of action steps constructed as a sum of two sets, let us call them A' and B', such that $A' \in \mathcal{ILL}^a(w)$ and $B' = means(w, X)$, where $X \in \mathcal{ILL}_s(w)$. Since B is a subset of A it contains only elements of A' and B'. Thus we know that all action steps from B are in an informal sense 'illegal'. However, we also have to be able to construct B in the way presented in Definition 2, that is a sum of an illegal action and means to an illegal state. That is not always possible. Let us consider the example presented on Figure 7.

Let $A = \{\langle w_1, w_2, a_1\rangle, \langle w_1, w_2, a_2\rangle\}$. By Definition 2 we have that $A \in \mathcal{ILL}^a_s(w)$. However $B = \{\langle w_1, w_2, a_1\rangle\}$, being a subset of A, is not an element of $\mathcal{ILL}^a_s(w)$ (note that $means(w_1, \{w_2\}) = \{\langle w_1, w_2, a_1\rangle, \langle w_1, w_2, a_2\rangle\}$).

In our opinion the fact that the counterpart of (1) does not hold in our model is an inessential peculiarity of $\mathcal{ILL}^a_s$ that does not undermine its role of the counterpart of $\mathcal{ILL}^a$ in the combined model. The intended meaning of $\mathcal{ILL}^a_s$ does not change: any member of $\mathcal{ILL}^a_s$ is a set of illegal transitions and behaving according to any of the transitions from such a set is prohibited. Thus, in general subsets of members of $\mathcal{ILL}^a_s$ should also be in $\mathcal{ILL}^a_s$. The only problem is that we cannot define such sets as B from the example within the language we use within the model.

2.3. Specific Norms in the Model

As we have mentioned in the introductory section, one of the main purposes of this paper is to derive specific norms from general *a-norms* and *s-norms*. Such specific norms describe what an agent should and should not do in a particular situation. We want to achieve that goal by finding the most precise

[13]Proofs are in the "Appendix".

general norm. We presented a similar solution for deontic action logic in [8] and now we extend it to the system including *s-norms*.

We shall express our most specific norm in the form of the largest set $\mathcal{ILL}^a_s(w)$. First let us notice that the set $\mathcal{REQ}^a_s(w)$ may be empty. Then the only norms to be considered are prohibitions. Moreover, in the case of non-empty $\mathcal{REQ}^a_s(w)$, due to the obligation economy principle (6) the most specific (thus the most informative) norm expressed as obligation has its prohibited counterpart. Due to conditions imposed on the set $\mathcal{ILL}^a_s(w)$ the largest element is also the sum of all the elements: $\bigcup \mathcal{ILL}^a_s(w)$.

It may happen that the choice of norms is such that $\bigcup \mathcal{ILL}^a_s(w)$ equals $\mathcal{S}tep$. In that case the system of norms in inconsistent in the sense that one cannot comply with it. In the opposite case $\mathcal{S}tep \backslash \bigcup \mathcal{ILL}^a_s(w)$ gives a complete recipe defining what an agent should do.

One of the benefits of the solution is the possibility of defining strong permission (or free choice permission—see [12,18,19]) within the model. As it is argued in [19] the notion is useful since, given a norm expressed with the use of strong permission, an agent can freely choose between the actions regulated by the norm and be sure that, whatever the choice is, it is legal and that no other norms or regulations have to be taken into account.

Since $\bigcup \mathcal{ILL}^a_s(w)$ collects all illegal steps, its complement, i.e., $\mathcal{S}tep \backslash \bigcup \mathcal{ILL}^a_s(w)$ collects all legal steps. Thus we can define the set $\mathcal{LEG}^a_s(w)$ similarly to $\mathcal{REQ}^a_s(w)$ and $\mathcal{ILL}^a_s(w)$, collecting norms that can be expressed as strong permissions.

$$\mathcal{LEG}^a_s(w) \triangleq 2^{\mathcal{S}tep \backslash \bigcup \mathcal{ILL}^a_s(w)} \tag{18}$$

The set $\mathcal{LEG}^a_s(w)$, defined as a powerset,[14] has the properties analogous to (1) and (2) as it is expected for strong permission; cf. [12,18].

If the set $\mathcal{REQ}^a_s(w)$ is not empty, then:

$$\mathcal{S}tep \backslash \bigcup \mathcal{ILL}^a_s(w) = \bigcap \mathcal{REQ}^a_s(w) \tag{19}$$

and we can define the set of legal steps as $\bigcap \mathcal{REQ}^a_s(w)$. The proof of (19) goes as follows:

Proof 1.

$$\mathcal{S}tep \backslash \bigcup \mathcal{ILL}^a_s(w) \subseteq \bigcap \mathcal{REQ}^a_s(w) \tag{20}$$

Assume that $X \notin \bigcap \mathcal{REQ}^a_s(w)$. *Since* $\mathcal{REQ}^a_s(w)$ *is not empty there exists* $A \in \mathcal{REQ}^a_s(w)$ *such that* $X \notin A$. *By* (6) $-A \in \mathcal{ILL}^a_s(w)$. *Further, because*

[14] Any powerset is an ideal in the algebraic sense.

$X \in -A$, $X \in \bigcup \mathcal{ILL}^a_s(w)$. Thus, $X \notin \mathcal{S}tep \backslash \bigcup \mathcal{ILL}^a_s(w)$.

$$\bigcap \mathcal{REQ}^a_s(w) \subseteq \mathcal{S}tep \backslash \bigcup \mathcal{ILL}^a_s(w) \tag{21}$$

By (4) $\bigcap \mathcal{REQ}^a_s(w) \in \mathcal{REQ}^a_s(w)$. If $\bigcap \mathcal{REQ}^a_s(w) = \emptyset$, then (21) is trivially true. Otherwise, assume indirectly that there exists X such that $X \in \bigcap \mathcal{REQ}^a_s(w)$ and $X \in \bigcup \mathcal{ILL}^a_s(w)$. So there exists $A \in \mathcal{ILL}^a_s(w)$ such that $X \in A$. Then, by (5) $\bigcap \mathcal{REQ}^a_s(w) \backslash X \in \mathcal{REQ}^a_s(w)$. That contradicts with X being a member of $\bigcap \mathcal{REQ}^a_s(w)$.

In our example from Figure 6 the set $\mathcal{REQ}^a_s(w) = \{\{\langle w_1, w_3, \text{safe\_action}\rangle\}\}$. Thus $\bigcap \mathcal{REQ}^a_s(w) = \{\langle w_1, w_3, \text{safe\_action}\rangle\}$ and $\bigcup \mathcal{ILL}^a_s(w)$ is its complement with respect to $\mathcal{S}tep$.

Let us finally notice one more property of the set $\bigcup \mathcal{ILL}^a_s(w)$. Namely, we can define it without the use of $\mathcal{ILL}^a_s(w)$, taking into account the following equation:

$$\bigcup \mathcal{ILL}^a_s(w) = \bigcup \mathcal{ILL}^a(w) \cup \bigcup means(w, \mathcal{ILL}_s(w)) \tag{22}$$

which follows from the fact that $\mathcal{ILL}^a_s(w)$ is a set of sums of all pairs of elements taken from $\mathcal{ILL}^a(w)$ and $means(w, \mathcal{ILL}_s(w))$. Thus, the sets $\mathcal{ILL}^a_s(w)$ and $\mathcal{REQ}^a_s(w)$ are not indispensable for defining specific norms.

3. Language for $\mathcal{DASF}$ and Its Interpretation

3.1. Language for $\mathcal{DASF}$

The language for $\mathcal{DASF}$ is defined in Backus–Naur notation in the following way:

$$\alpha \ ::= \ a_i \mid \mathbf{0} \mid \mathbf{1} \mid \overline{\alpha} \mid \alpha \sqcup \alpha \mid \alpha \sqcap \alpha \tag{23}$$

$$\varphi \ ::= \ p_i \mid \bot \mid \top \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \tag{24}$$

$$\psi \ ::= \ \alpha = \alpha \mid \mathsf{F}_a(\alpha) \mid \mathsf{O}_a(\alpha) \mid \mathsf{F}_s(\varphi) \mid \mathsf{O}_s(\varphi) \mid \neg\psi \mid \psi \wedge \psi \tag{25}$$

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, '$\overline{\alpha}$'—not α (complement of α), '$\alpha \sqcup \beta$'—α or β (a free choice between α and β); '$\alpha \sqcap \beta$'—α and β (parallel execution of α and β); '$\alpha = \beta$' means that α is identical with β; '$\mathsf{F}_a(\alpha)$'—α is forbidden, '$\mathsf{O}_a(\alpha)$'—α is obligatory; p_i belongs to a set of atomic propositions Atm, '$\bot$' and '$\top$' represent falsehood and truth, respectively; '$\mathsf{O}_s(\varphi)$'—the state of affairs φ is obligatory; $\mathsf{F}_s(\varphi)$'—the state of affairs φ is forbidden.

For a fixed Act_0, by Act we shall understand a set of formulas defined by (23). Obviously $Act_0 \subseteq Act$. We shall also use '$\rightarrow$' and '$\equiv$' as implication and equivalence respectively, defined in the usual way. Let us notice that the language is protected from iteration of deontic state operators.[15]

It should be also stressed that in the language of our logic there are neither PDL-like nor STIT-like operators expressing execution of actions or agency ('sees to it that'). This is because we intended to stay on the level of deontic operators and restrict our research here to understanding the mutual dependency between *a-norms* and *s-norms*. The PDL-like and STIT-like operators can be easily added to our framework in the similar way as shown for instance by Sergot in [13,14]. The key point of our and Sergot's approaches (Sergot's and our models are very similar) is that we treat the models as more fundamental for analyses of deontic issues. To the language of logic are added only these operators that are useful for capturing some aspect of the model that are currently under investigation.

3.2. Interpretation for Actions and Satisfaction Conditions for Deontic Action Operators

$\mathcal{I}^a$:$Act \longrightarrow 2^{\mathcal{S}tep}$ is an interpretation function for $\mathcal{DAF}$ defined as follows:

$$\mathcal{I}^a(a_i) \subseteq \mathcal{S}tep, \text{ for } a_i \in Act_0 \tag{26}$$

$$\mathcal{I}^a(\mathbf{0}) = \emptyset \quad \mathcal{I}^a(\mathbf{1}) = \mathcal{S}tep \tag{27}$$

$$\mathcal{I}^a(\alpha \sqcup \beta) = \mathcal{I}^a(\alpha) \cup \mathcal{I}^a(\beta) \tag{28}$$

$$\mathcal{I}^a(\alpha \sqcap \beta) = \mathcal{I}^a(\alpha) \cap \mathcal{I}^a(\beta) \tag{29}$$

$$\mathcal{I}^a(\overline{\alpha}) = \mathcal{S}tep \backslash \mathcal{I}^a(\alpha) \tag{30}$$

Thus, every action generator is interpreted as a set of labelled transitions, the impossible action has no transitions, the universal action is interpreted as a set of all possible transitions, operations '$\sqcup$', '$\sqcap$' between actions and '$^-$' on a single action are interpreted as set-theoretical operations on interpretations of actions.

For fixed $a_i \in Act_0$ we also assume that if an action step with a label $e \in \mathcal{E}$ belongs to $\mathcal{I}^a(a_i)$, then all action steps with that label e are elements of $\mathcal{I}^a(a_i)$, formally:

$$\forall e\ (\exists w_1, w_2\ \langle w_1, w_2, e\rangle \in \mathcal{I}^a(a_i) \Longrightarrow \forall w_1', w_2' \langle w_1', w_2', e\rangle \in \mathcal{I}^a(a_i)) \tag{31}$$

[15] Iterated deontic state formulas are often interpreted as norms of higher order. Because *a-norms* are never higher-order norms, it makes sense (at least it is not controversial) to link them with *s-norms* of the same level. That is the reason we have restricted our *s-norms* language.

$\mathcal{I}^a$ is an interpretation of actions insensitive to their preconditions and takes into account all the possible executions of actions in all the states in which they can be executed. To make our interpretation related to a particular state we introduce $\mathcal{I}^a(w, \alpha)$ that is a local interpretation of action (relativized to situation w) and define it as follows:

$$\mathcal{I}^a(w, \alpha) = \mathcal{I}^a(\alpha) \cap exe(w), \tag{32}$$

where $exe(w) \triangleq \{\langle w, w', e\rangle : \langle w, w', e\rangle \in \mathcal{S}tep\}$ is a set of all action steps executable in state w.

Satisfaction conditions for the action formulas in any model $\mathcal{M} = \langle \mathcal{DAF}, \mathcal{I}^a \rangle$ are defined below:

$$\begin{array}{lcl} \mathcal{M}, w \models \mathsf{F}_a(\alpha) & \iff & \mathcal{I}^a(w, \alpha) \in \mathcal{ILL}^a(w) \\ \mathcal{M}, w \models \mathsf{O}_a(\alpha) & \iff & \mathcal{I}^a(w, \alpha) \in \mathcal{REQ}^a(w) \\ \mathcal{M}, w \models \alpha = \beta & \iff & \mathcal{I}^a(\alpha) = \mathcal{I}^a(\beta) \end{array}$$

3.3. Interpretation for Propositions and Deontic State Operators

$v : Atm \longrightarrow 2^{\mathcal{W}}$ is a standard valuation function that assigns a subset $v(p_i)$ of $\mathcal{W}$ to each proposition in Atm. We shall think of $v(p_i)$ as a semantical representation of proposition p_i in the model, i.e., a set of states in $\mathcal{W}$ where p is true (takes place).

Satisfaction conditions for the state formulas in any model $\mathcal{M} = \langle \mathcal{DSF}, v \rangle$ are defined below:

$$\mathcal{M}, w \models \mathsf{O}_s(\varphi) \iff \|\varphi\|^{\mathcal{M}} \in \mathcal{REQ}_s(w)$$
$$\mathcal{M}, w \models \mathsf{F}_s(\varphi) \iff \|\varphi\|^{\mathcal{M}} \in \mathcal{ILL}_s(w)$$

$\|\varphi\|^{\mathcal{M}}$ is a *truth set* of the sentence φ in the model $\mathcal{M}$, i.e., a set of states at which φ is true. Formally[16]: $\|\varphi\|^{\mathcal{M}} = \{w \in \mathcal{W} : \mathcal{M}, w \models \varphi\}$.

3.4. A Bridge Between Deontic Actions and States

Now we introduce deontic operators, O and F, combing actions and results. Both operators have two arguments—an action and a formula being a result of the action. We shall read them in natural language and understand them intuitively as follows:

[16]The following facts about a truth set are known: $\|p_i\|^{\mathcal{M}} = v(p_i)$, for every $p_i \in Atm$; $\|\top\|^{\mathcal{M}} = \mathcal{W}$; $\|\neg\varphi\|^{\mathcal{M}} = \mathcal{W} \backslash \|\varphi\|^{\mathcal{M}} = -\|\varphi\|^{\mathcal{M}}$; $\|\varphi \wedge \psi\|^{\mathcal{M}} = \|\varphi\|^{\mathcal{M}} \cap \|\psi\|^{\mathcal{M}}$; $\|\varphi \vee \psi\|^{\mathcal{M}} = \|\varphi\|^{\mathcal{M}} \cup \|\psi\|^{\mathcal{M}}$.

- '$\mathsf{O}(\alpha, \varphi)$'—it is obligatory to execute α in such a way that φ. For '$\mathsf{O}(\alpha, \varphi)$' to be true one of the three cases should take place (i) α and φ are both obligatory, (ii) α is obligatory and $\neg\varphi$ is prohibited or (iii) φ is obligatory and $\overline{\alpha}$ is prohibited. In other words both $\overline{\alpha}$ and $\neg\varphi$ are forbidden and at least one of the two: α or φ is obligatory (compare Definition 1 and Observation 1 following it).
- '$\mathsf{F}(\alpha, \varphi)$'—it is forbidden to execute α or bring about φ. For a particular behavior to be forbidden it is enough that one out of the two conditions is fulfilled. However, for '$\mathsf{F}(\alpha, \varphi)$' to be true both α and φ should be forbidden.

Let us come back to our contract scenario (see Example 1). The appropriate language is based on the following sets of action generators and atomic propositions: $Act_0 = \{\text{safe\_action}\}$ and $\text{Atm} = \{\text{works\_done}, \text{accident\_happened}\}$. Let unsafe_action be the complement of safe_action, i.e., $\text{unsafe\_action} = \overline{\text{safe\_action}}$. Within this language we can formulate the following example of binary obligation: $\mathsf{O}(\text{safe\_action}, \text{works\_done})$. It is obligatory to execute safe_action (undertake only actions that are safe) in such a way that works_done is the case (getting to a situation in which planned works are done). At the same time we can say that it is forbidden to execute unsafe_action (no matter what its results will be) or bring about accident_happen (no matter by means of which action):

$$\mathsf{F}(\text{unsafe\_action}, \text{accident\_happened}).$$

Similarly we can say that it is forbidden to execute unsafe_action or bring about ¬works_done. In other words it is forbidden that not all actions are safe or works are not done: $\mathsf{F}(\text{unsafe\_action}, \neg\text{works\_done})$.

3.4.1. Formal Definitions for O and F.

Formally we can define the new operators in the following way:

$$\mathsf{O}(\alpha, \varphi) \triangleq (\mathsf{O}_a(\alpha) \wedge \mathsf{F}_s(\neg\varphi)) \vee (\mathsf{O}_s(\varphi) \wedge \mathsf{F}_a(\overline{\alpha})) \tag{33}$$

$$\mathsf{F}(\alpha, \varphi) \triangleq \mathsf{F}_a(\alpha) \wedge \mathsf{F}_s(\varphi) \tag{34}$$

3.4.2. Satisfaction Conditions for O and F.

Let us now turn to the satisfaction conditions for O and F. The same set of steps can be obtained as an interpretation of different 'action-proposition' pairs taken as arguments of operators O and F. Moreover, in the case of O we can have the same set of steps for pairs α, φ and β, φ even if $\mathcal{I}^a(\alpha) \neq \mathcal{I}^a(\beta)$ (the same holds for pairs α, φ and α, ψ). That is because we postulate that binary obligations emerge as results of trimming obligatory actions by means of their forbidden results or selecting from permitted actions those which lead to obligatory results.

For example let us take the following binary obligation in the language from our Example 1: $\mathsf{O}(\text{safe\_action}, \text{work\_done} \vee \text{accident\_happened})$. Intuitively we can understand '$\text{work\_done} \vee \text{accident\_happened}$' as meaning that unless an accident happens the work is to be completed. The combination of actions and states representing arguments of that formula in the model gives us the singleton set $\{\langle w_1, w_3, \text{safe\_action}\rangle\}$. We have already checked that this set belongs to $\mathcal{REQ}^a_s$ from the example. However, neither argument of the binary obligation operator is obligatory by itself (as an obligatory action or an obligatory state). Thus, the binary obligation cannot be valid. This fact shows us that binary obligation O, as defined in the paper, being closely related to $\mathcal{REQ}^a_s$ cannot be characterized only by it.

In contrast, a two-argument prohibition holds for any combination of an action and a proposition whose interpretation is a subset of the interpretation of a forbidden pair.

Observation 2. *The above remarks can be formalized with the use of the following satisfaction conditions:*

$$\mathcal{M}, w \models \mathsf{O}(\alpha, \varphi) \iff (\mathcal{I}^a(w, \alpha) \in \mathcal{REQ}^a(w) \textit{ or } \|\varphi\|^{\mathcal{M}} \in \mathcal{REQ}_s(w)) \;\&\; \mathcal{I}^a(w, \alpha) \cap \textit{means}(w, \|\varphi\|^{\mathcal{M}}) \in \mathcal{REQ}^a_s(w)$$

$$\mathcal{M}, w \models \mathsf{F}(\alpha, \varphi) \iff (\mathcal{I}^a(w, \alpha) \cup \textit{means}(w, \|\varphi\|^{\mathcal{M}})) \in \mathcal{ILL}^a_s(w)$$

It is easy to check that the definitions of the operators and models make the conditions in the observation above fulfilled.

4. Logics for Deontic Actions, Deontic States and Their Combination

4.1. Logics for Deontic Actions and Deontic States

Deontic action logic is expressed in the language defined by conditions (23) and (25) without O_s and F_s operators. Its axiomatization corresponding with the $\mathcal{DAF}$ (see Section 2.1) comes from [18]. It consists of the rule Modus Ponens of the usual form, the rule of extensionality: if $\alpha = \beta$ and $\Phi(\alpha)$, then $\Phi(\beta)$ ($\Phi(\alpha)$ stands for any formula in which the action name α appears) and the following axioms:

$$\text{Boolean algebra for actions from } \mathit{Act}. \tag{35}$$

$$\mathsf{F}_a(\alpha \sqcup \beta) \equiv \mathsf{F}_a(\alpha) \wedge \mathsf{F}_a(\beta) \tag{36}$$

$$\mathsf{F}_a(\mathbf{0}) \tag{37}$$

$$\mathsf{O}_a(\alpha) \wedge \mathsf{O}_a(\beta) \to \mathsf{O}_a(\alpha \sqcap \beta) \tag{38}$$

$$\mathsf{O}_a(\alpha) \to \mathsf{F}_a(\overline{\alpha}) \tag{39}$$

$$\mathsf{O}_a(\alpha) \wedge \mathsf{F}_a(\beta) \to \mathsf{O}_a(\alpha \sqcap \overline{\beta}) \tag{40}$$

Deontic state logic is expressed in the language defined by conditions (24) and (25) without O_a and F_a operators. Its axiomatization corresponding with the $\mathcal{DSF}$ (see Section 2.1) is analogous to the axioms above (of course extensionality concerns here equivalent formulas and Boolean algebra is substituted by classical propositional calculus).

4.2. Tautologies for Binary Deontic Operators

The formulas below are theses of the combined deontic logic. They correspond to the axioms of deontic action (state) logic presented above.

$$\mathsf{F}(\alpha \sqcup \beta, \varphi) \equiv \mathsf{F}(\alpha, \varphi) \wedge \mathsf{F}(\beta, \varphi) \tag{41}$$

$$\mathsf{F}(\mathbf{0}, \varphi) \tag{42}$$

$$\mathsf{F}(\alpha, \varphi \vee \psi) \equiv \mathsf{F}(\alpha, \varphi) \wedge \mathsf{F}(\alpha, \psi) \tag{43}$$

$$\mathsf{F}(\alpha, \bot) \tag{44}$$

$$\mathsf{O}(\alpha, \varphi) \wedge \mathsf{O}(\beta, \varphi) \to \mathsf{O}(\alpha \sqcap \beta, \varphi) \tag{45}$$

$$\mathsf{O}(\alpha, \varphi) \wedge \mathsf{O}(\alpha, \psi) \to \mathsf{O}(\alpha, \varphi \wedge \psi) \tag{46}$$

$$\mathsf{O}(\alpha, \varphi) \wedge \mathsf{F}(\beta, \varphi) \to \mathsf{O}(\alpha \sqcap \overline{\beta}, \varphi) \tag{47}$$

$$\mathsf{O}(\alpha, \varphi) \wedge \mathsf{F}(\alpha, \psi) \to \mathsf{O}(\alpha, \varphi \wedge \neg\psi) \tag{48}$$

$$\mathsf{O}(\alpha, \varphi) \to \mathsf{F}(\overline{\alpha}, \neg\varphi) \tag{49}$$

Our forbiddance operator is strong; as such it is formally similar to van der Meyden's strong permission π (see [19]). So, by analogy, our F operator satisfies the same axiom schemas (see π3. and π5. in [19]).

4.3. Bridging Formulas

Some bridging formulas, connecting the defined operators with the primitive ones, follow immediately from definitions (33) and (34).

$$\mathsf{O}_a(\alpha) \wedge \mathsf{O}_s(\varphi) \to \mathsf{O}(\alpha, \varphi) \tag{50}$$

$$\mathsf{O}(\alpha, \varphi) \to \mathsf{F}_a(\overline{\alpha}) \tag{51}$$

$$\mathsf{O}(\alpha, \varphi) \to \mathsf{F}_s(\neg\varphi) \tag{52}$$

$$\mathsf{O}(\alpha, \varphi) \to \mathsf{O}_a(\alpha) \vee \mathsf{O}_s(\varphi) \tag{53}$$

More interesting relations between binary and unary deontic operators can be formulated for specific systems of norms. Let us, for instance, consider a system in which there are no *a-norm*s. No action should then be obligatory or forbidden. However, we need to take into account that $\mathbf{0}$ is always forbidden by axiom (37). Thus in the normative system of our interest we have the following axiom:

$$\mathsf{F}_a(\alpha) \rightarrow \alpha = \mathbf{0} \tag{54}$$

and its equivalent: $\mathsf{F}_a(\alpha) \equiv \alpha = \mathbf{0}$. It follows from (54) that only the universal action $\mathbf{1}$ can be obligatory in such a system, formally

$$\mathsf{O}_a(\alpha) \rightarrow \alpha = \mathbf{1} \tag{55}$$

However, we may also require the stronger version of (55):

$$\neg \mathsf{O}_a(\alpha) \tag{56}$$

If *a-norm*s are absent, then all norms are *s-norm*s. We can express this idea by the following formula:

$$((\mathsf{F}_a(\beta_1) \rightarrow \beta_1 = \mathbf{0}) \wedge \neg \mathsf{O}_a(\beta_2)) \rightarrow (\mathsf{F}(\alpha, \varphi) \equiv \mathsf{F}_s(\varphi)) \wedge (\mathsf{O}(\alpha, \varphi) \equiv \mathsf{O}_s(\varphi)) \tag{57}$$

In the weaker version we have:

$$(\mathsf{F}_a(\beta) \rightarrow \beta = \mathbf{0}) \rightarrow (\mathsf{F}(\alpha, \varphi) \equiv \mathsf{F}_s(\varphi)) \tag{58}$$

Similarly, we can define a system in which there are no *s-norm*s.

4.4. Specific Norms in the Logic

4.4.1. Definition of the Most General (i.e., the weakest) Strong Prohibition. To introduce specific norms concerning prohibition into the logic we shall use an additional operator of the most general forbiddance '$\mathsf{F}^{\#}$'. It is not possible to define it within the language, so we use the following metalanguage definition[17]:

$$\mathsf{F}^{\#}(\alpha, \varphi) \triangleq \mathsf{F}(\alpha, \varphi) \ \& \ \forall \beta, \psi \ (\mathsf{F}(\beta, \psi) \Longrightarrow \beta \sqsubseteq \alpha \ \& \ \varphi \rightarrow \psi) \tag{59}$$

$\mathsf{F}^{\#}(\alpha, \varphi)$ indicates a unique pair of arguments defining the space of forbidden actions and propositions. Everything that is not described by that pair is permitted.

[17]Formula $\beta \sqsubseteq \alpha$ expresses the fact that β is more specific then α. Formally

$$\beta \sqsubseteq \alpha \triangleq \alpha \sqcap \beta = \beta.$$

4.4.2. Axioms for the Weakest Strong Prohibition. '$\mathsf{F}^{\#}$' can be also introduced axiomatically provided there is the operator of strong permission (for which we shall use the symbol P). A Segerberg-style axiomatization of 'P' can be formulated as follows:

$$\mathsf{P}(\alpha \sqcup \beta, \varphi) \equiv \mathsf{P}(\alpha, \varphi) \wedge \mathsf{P}(\beta, \varphi) \tag{60}$$

$$\mathsf{P}(\mathbf{0}, \varphi) \tag{61}$$

$$\mathsf{P}(\alpha, \varphi \vee \psi) \equiv \mathsf{P}(\alpha, \varphi) \wedge \mathsf{P}(\alpha, \psi) \tag{62}$$

$$\mathsf{P}(\alpha, \bot) \tag{63}$$

$$\mathsf{P}(\alpha, \varphi) \wedge \mathsf{F}(\alpha, \varphi) \rightarrow \alpha = \mathbf{0} \vee (\varphi \equiv \bot) \tag{64}$$

Formulas (60)–(63) are analogous to formulas (41) and (44) for prohibition. Formula (64) states that no action or state should be at the same time forbidden and strongly permitted. Formulas (60)–(64) correspond to axioms defining strong permission in [12,18].

Let us notice that strong permission has some properties that are regarded as permission paradoxes. However, they are paradoxical only when we want to connect them with weak permission (lack of prohibition) usually used in natural language. In the context of strong permission (free choice permission) they are quite natural.

Then the weakest strong prohibition can be characterized by two postulates expressing the necessary and sufficient conditions for its occurrence:

$$\mathsf{F}^{\#}(\alpha, \varphi) \rightarrow \mathsf{P}(\overline{\alpha}, \neg\varphi) \tag{65}$$

$$\mathsf{O}(\beta, \psi) \wedge \neg\mathsf{O}_a(\mathbf{0}) \rightarrow (\mathsf{O}(\alpha, \varphi) \wedge \mathsf{P}(\alpha, \varphi) \rightarrow \mathsf{F}^{\#}(\overline{\alpha}, \neg\varphi)) \tag{66}$$

The first postulate is a necessary condition for the weakest strong prohibition establishing dependence between it and the strong permission. The second one, the sufficient condition, states that if there exists any obligation in the normative system and obligations are consistent, then the fact that it is at the same time obligatory and strongly permitted to execute α in such a way that φ is its result implies that it is forbidden (in the $\mathsf{F}^{\#}$-sense) to execute any action that realizes the complement of α or brings about $\neg\varphi$ as a result.

It is worth noting that formula (66) is a non-quantifier version of the formula below:

$$(\exists\beta, \psi\ \mathsf{O}(\beta, \psi)) \wedge \neg\mathsf{O}_a(\mathbf{0}) \Longrightarrow \mathsf{O}(\alpha, \varphi) \wedge \mathsf{P}(\alpha, \varphi) \rightarrow \mathsf{F}^{\#}(\overline{\alpha}, \neg\varphi)$$

Strong prohibition can be also used to obtain the most specific obligation. Thus, provided there exists any obligation in the normative system and obligations are consistent, the weakest strong prohibition implies the most specific obligation:

$$\mathsf{O}(\beta,\psi) \wedge \neg \mathsf{O}_a(\mathbf{0}) \rightarrow (\mathsf{F}^{\#}(\alpha,\varphi) \rightarrow \mathsf{O}(\overline{\alpha},\neg\varphi)) \tag{67}$$

A similar intuition about the relationship between strong permission and obligation understood as the most specific norm was presented in [1,5,11].

4.4.3. Specific *a-norm***s and** *s-norm***s.** The most general prohibition in a normative system which takes into consideration *a-norm*s and *s-norm*s can also be introduced in a different way. Let us first recall formula (22) in which a model-theoretic counterpart of the most general prohibition ($\mathcal{ILL}_s^a$) is defined on the basis of sets of transitions representing prohibited actions ($\mathcal{ILL}^a$) and prohibited states ($\mathcal{ILL}_s$). We can now reproduce the same reasoning on the level of logic.

The most general prohibited action and the most general prohibited proposition can be defined as follows:

$$\mathsf{F}_a^{\#}(\alpha) \triangleq \mathsf{F}_a(\alpha) \;\&\; \forall\beta\; (\mathsf{F}_a(\beta) \Longrightarrow \beta \sqsubseteq \alpha) \tag{68}$$

$$\mathsf{F}_s^{\#}(\varphi) \triangleq \mathsf{F}_s(\varphi) \;\&\; \forall\psi\; (\mathsf{F}_s(\psi) \Longrightarrow \varphi \rightarrow \psi) \tag{69}$$

With those definitions we can easily see that the following equation which establishes a relation between the most general prohibition in general and the most general prohibited actions and states, holds:

$$\mathsf{F}^{\#}(\alpha,\varphi) \equiv (\mathsf{F}_a^{\#}(\alpha) \wedge \mathsf{F}_s^{\#}(\varphi)) \tag{70}$$

5. Conclusions and Further Work

This paper presents a framework that enables reasoning about action and state norms in a unified manner. Its main achievements are: (1) the model-theoretic structure for action and state obligations and prohibitions for general norms, (2) the notion of specific norms, (3) the binary operators for obligation and prohibition introduced into the language of logic and their meaning defined on the basis of two separate simple logics (deontic action logic and deontic state logic). The defined operators have been proved to be adequate with respect to the model.

Several ideas can be further developed. Since in our theory we take into account only one-step actions (transitions), we are interested in extending it to sequences of actions. As we have shown in [9] the move from one-step actions to their sequences is far from trivial.

Other important issues of action theory in general, and deontic action logic in particular, are agency and agents. We do not introduce agents into our formalism explicitly, but we tacitly assume that actions are carried out by them. We can extend the system by incorporating agents into names of basic actions in the way that is analogical to [13,14], where agents' names are parts of transition atoms. That would enable us to extend the presented solutions in a multi-agent system direction.

Acknowledgements. This research is supported by the National Science Centre of Poland (UMO-2015/17/B/HS1/02569). The authors would like to thank Marek Sergot, Xavier Parent, Marek Piechowiak and anonymous reviewers for their remarks on the previous versions of the paper.

Appendix: Proofs of the Properties of $\mathcal{DASF}$

The counterpart of conditions (2)–(6) are the following formulas (71), (73), (74), (76) and (78) respectively.

$$A \in \mathcal{ILL}^a_s(w) \;\&\; B \in \mathcal{ILL}^a_s(w) \Longrightarrow A \cup B \in \mathcal{ILL}^a_s(w) \tag{71}$$

1.	$A \in \mathcal{ILL}^a_s(w)$	Assumption
2.	$B \in \mathcal{ILL}^a_s(w)$	Assumption
3.	$\exists_{A',X} A = A' \cup means(w, X) \;\&$ $A' \in \mathcal{ILL}^a(w) \;\&\; X \in \mathcal{ILL}_s(w)$	Definition 2, 1
4.	$\exists_{B',Y} B = B' \cup means(w, Y) \;\&$ $B' \in \mathcal{ILL}^a(w) \;\&\; Y \in \mathcal{ILL}_s(w)$	Definition 2, 2
5.	$A' \cup B' \in \mathcal{ILL}^a(w)$	(2), 3, 4
6.	$X \cup Y \in \mathcal{ILL}_s(w)$	(2), 3, 4
7.	$(A' \cup B') \cup means(w, X \cup Y) \in \mathcal{ILL}^a_s(w)$	Definition 2, 5, 6
	$A \cup B \in \mathcal{ILL}^a_s(w)$	(72), 7

For the proof we used the following equation:

$$\begin{aligned}(A' \cup B') \cup means(w, X \cup Y) &= (A' \cup B') \cup means(w, X) \cup means(w, Y) \\ &= (A' \cup means(w, X)) \cup (B' \cup means(w, Y)) \\ &= A \cup B \end{aligned} \tag{72}$$

$$\emptyset \in \mathcal{ILL}^a_s(w) \tag{73}$$

By (3) we have that $\emptyset \in \mathcal{ILL}^a(w)$ and $\emptyset \in \mathcal{ILL}_s(w)$. By means property we have also that $means(w, \emptyset) = \emptyset$. Then by Definition 2 we obtain (73).

$$A \in \mathcal{REQ}^a_s(w) \;\&\; B \in \mathcal{REQ}^a_s(w) \Longrightarrow A \cap B \in \mathcal{REQ}^a_s(w) \tag{74}$$

1.	$A \in \mathcal{REQ}^a_s(w)$	Assumption
2.	$B \in \mathcal{REQ}^a_s(w)$	Assumption
3.	$\exists_{A',X} A = A' \cap means(w, X)$ & $(A' \in \mathcal{REQ}^a(w)$ & $-X \in \mathcal{ILL}_s(w)$ or $-A' \in \mathcal{ILL}^a(w)$ & $X \in \mathcal{REQ}_s(w))$	Definition 1, 1
4.	$\exists_{B',Y} B = B' \cap means(w, Y)$& $B' \in \mathcal{REQ}^a(w)$ & $-Y \in \mathcal{ILL}_s(w)$ or $-B' \in \mathcal{ILL}^a(w)$ & $Y \in \mathcal{REQ}_s(w)$	Definition 1, 2
3a.	$A' \in \mathcal{REQ}^a(w)$ & $-X \in \mathcal{ILL}_s(w)$	3: case a
4a.	$B' \in \mathcal{REQ}^a(w)$ & $-Y \in \mathcal{ILL}_s(w)$	4: case a
5.	$A' \cap B' \in \mathcal{REQ}^a(w)$	(4), 3a, 4a
6.	$-X \cup -Y \in \mathcal{ILL}_s(w)$	(2), 3a, 4a
7.	$(A' \cap B') \cap -means(w, -X \cup -Y) \in \mathcal{REQ}^a(w)$	Definition 1, 5, 6
	$A \cap B \in \mathcal{REQ}^a_s(w)$	(75), 7

$$\begin{aligned}(A' \cap B') \cap -means(w, -X \cup -Y) &= (A' \cap B') \cap means - (w, -X \cup -Y) \\ &= (A' \cap B') \cap means(w, X \cap Y) \\ &= (A' \cap B') \cap means(w, X) \cap means(w, Y) \\ &= A \cap B \end{aligned} \tag{75}$$

The complete proof is a proof by cases. Steps 3 and 4 of the proof are disjunctions. Thus, there are four cases to be considered. We present one of them starting from steps 3a and 4a, the remaining three cases are analogous.

$$A \in \mathcal{REQ}^a_s(w) \text{ and } B \in \mathcal{ILL}^a_s(w) \Longrightarrow A \cap -B \in \mathcal{REQ}^a_s(w) \tag{76}$$

1.	$A \in \mathcal{REQ}^a_s(w)$	Assumption
2.	$B \in \mathcal{ILL}^a_s(w)$	Assumption
3.	$\exists_{A',X} A = A' \cap means(w, X)$ & $(A' \in \mathcal{REQ}^a(w)$ & $-X \in \mathcal{ILL}_s(w)$ or $-A' \in \mathcal{ILL}^a(w)$ & $X \in \mathcal{REQ}_s(w))$	Definition 1, 1
4.	$\exists_{B',Y} B = B' \cup means(w, Y)$ & $B' \in \mathcal{ILL}^a(w)$ & $Y \in \mathcal{ILL}_s(w)$	Definition 2, 2
3a.	$A' \in \mathcal{REQ}^a(w)$ & $-X \in \mathcal{ILL}_s(w)$	3: case a
5.	$A' \cap -B' \in \mathcal{REQ}^a$	(5), 3a, 4
6.	$-X \cup Y \in \mathcal{ILL}_s$	(2), 3a, 4
7.	$(A' \cap -B') \cap -means(w, -X \cup Y) \in \mathcal{REQ}^a(w)$	Definition 1, 5, 6
	$A \cap -B \in \mathcal{REQ}^a_s(w)$	(77), 7

For the proof we used the following equation:

$$\begin{aligned}
&(A' \cap -B') \cap -means(w, -X \cup Y) \\
&\quad = (A' \cap -B') \cap means - (w, -X \cup Y) \\
&\quad = (A' \cap -B') \cap means(w, X \cap -Y) \\
&\quad = (A' \cap -B') \cap means(w, X) \cap means(w, -Y) = A \cap -B
\end{aligned} \tag{77}$$

As in the previous proof this one is a proof by cases with respect to step 3. The remaining case is analogous.

$$A \in \mathcal{REQ}^a_s(w) \Longrightarrow -A \in \mathcal{ILL}^a_s(w) \tag{78}$$

1.	$A \in \mathcal{REQ}^a_s(w)$	Assumption
2.	$\exists_{A',X} A = A' \cap means(w, X)$ & $(A' \in \mathcal{REQ}^a(w)$ & $-X \in \mathcal{ILL}_s(w)$ or $-A' \in \mathcal{ILL}^a(w)$ & $X \in \mathcal{REQ}_s(w))$	Definition 1, 1
2a.	$A' \in \mathcal{REQ}^a(w)$ & $-X \in \mathcal{ILL}_s(w)$	2: case a
3.	$-A' \in \mathcal{ILL}^a(w)$ & $-X \in \mathcal{ILL}_s(w)$	(6), 3
4.	$-A' \cup means(w, -X) \in \mathcal{ILL}^a_s(w)$	Definition 2
	$-A \in \mathcal{ILL}^a_s(w)$	4

Again, the remaining case of the proof is omitted as analogous to the presented one.

References

[1] Anglberger, A. J. J., N. Gratzl, and O. Roy, Obligation, free choice, and the logic of weakest permissions. *Rew. Symb. Logic* 8(4):807–827, 2015.

[2] Atienza, M., and J. Ruiz Manero, *A theory of legal sentences*, Kluwer, 1998.

[3] Broersen, J., D. Gabbay, and L. van der Torre, Changing norms is changing obligation change, in T. Agotnes, J. Brosen, and D. Elgesem, (eds.), *Lectures on Logic and Computation*, vol. 7393 of *Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, 2012, pp. 199–214.

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

[5] Czelakowski, J., Action and deontology, in E. Ejerhed and S. Lindström, (eds.), *Logic, action, and cognition: essays in philosophical logic*, Trends in Logic, Kluwer Academic, 1997, pp. 47–88.

[6] d'Altan, P., J.-J.Ch. Meyer, and R. J. Wieringa, (eds.), An integrated framework for ought-to-be and ought-to-do constraints. *Artificial Intelligence and Law* 4(2):77–111, 1996.

[7] Gabbay, D., L. Gammaitoni, and X. Sun, The paradoxes of permission: an action based solution. *Journal of Applied Logic* 12(2):179–191, 2014.

[8] Kulicki, T., Two faces of obligation, in A. Brozek, J. Jadacki, and B. Zarnic (eds.), *Theory of Imperatives from Different Points of View*, Logic, Methodology and Philosophy of Science at Warsaw University 7. Wydawnictwo Naukowe Semper, 2013.

[9] Kulicki, P., and R. Trypuz, Completely and partially executable sequences of actions in deontic context. *Synthese* 192(4):1117–1138, 2015.

[10] Pattaro, E., *The Law and The Right*, vol. 1 of *A Treatise of Legal Philosophy and General Jurisprudence*, Springer, 2012.

[11] Roy, O., A. J. J. Anglberger, and N. Gratzl, The Logic of Obligation as Weakest Permission, in T., Ågotnes, J. Broersen, and D. Elgesem, (eds.), *DEON 2012*, vol 7393 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 139–150.

[12] Segerberg, K., A deontic logic of action. *Studia Logica* 41:269–282, 1982.

[13] Sergot, M., *The logic of unwitting collective agency.* Technical Report 2008/6, Department of Computing, Imperial College London, 2008.

[14] Sergot, M., Some examples formulated in a 'seeing to it that' logic: Illustrations, observations, problems, in T. Müller, (ed.), *Nuel Belnap on Indeterminism and Free Action*, Springer, 2014, pp. 223–256.

[15] Sergot, M. J., and R. Craven, The Deontic Component of Action Language nC+, in *Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006, Proceedings*, 2006, pp. 222–237.

[16] Trypuz, R., and P. Kulicki, A systematics of deontic action logics based on boolean algebra. *Logic and Logical Philosophy* 18:263–279, 2009.

[17] Trypuz, R., and P. Kulicki, Deontic logic of actions and states, in F. Carini, D. Grossi, J. Mehus, and X. Parent, (eds.), *Deontic Logic and Normative System*, LNAI 8554, Springer, 2014, pp. 258–272.

[18] Trypuz, R., and P. Kulicki, On deontic action logics based on boolean algebra. *Journal of Logic Computation* 25(5):1241–1260, 2015.

[19] van der Meyden, R., The Dynamic Logic of Permission. *Journal of Logic and Computation* 6(3):465–479, 1996.

[20] von Wright, G. H., *Norm and Action: A Logical Inquiry*, Routledge & Kegan Paul, 1963.

[21] Ziembinski, Z., *Practical Logic*, D. Reidel Pub. Co, Dordrecht, 1976.

P. Kulicki, R. Trypuz
Faculty of Philosophy
John Paul II Catholic University of Lublin
Al. Raclawickie 14
Lublin
Poland
trypuz@kul.pl

P. Kulicki
kulicki@kul.pl