Allowed, or enabled, that is the question

Giovanni Sileno [1]

Informatics Institute, University of Amsterdam
Science Park 900, 1098 XH Amsterdam, the Netherlands

Matteo Pascucci

Institute of Philosophy of the Slovak Academy of Sciences, v.v.i.
Klemensova 19, 811 09 Bratislava, Slovakia

Réka Markovich

Department of Computer Science, University of Luxembourg
2 Avenue de l'Université, 4365 Esch-Belval Esch-sur-Alzette, Luxembourg

Abstract

The formal analysis of normative systems has traditionally focused on their deontic dimension rather than on their potestative dimension; yet, a growing amount of works aims at shedding light on the notion of power, its norm changing potential and its general interactions with deontic concepts. The present article contributes to this line of inquiry by adopting the following perspective: a normative system can be metaphorically seen as an agent that allocates abilities (powers) in order to promote the fulfillment of certain desires (deontic directives), and in doing so regulates its behavioural domain. Our analysis emphasizes the instrumental nature of power, while clarifying the distinction between 'being allowed' and 'being enabled' and unveiling new patterns of interaction between deontic and potestative concepts. Operationally, we formulate this framework in terms of conditional rules, and provide a corresponding logic programming (ASP) implementation.

Keywords: Normative Systems; Permission; Power; Instrumental Reasoning; Answer Set Programming

1 Introduction

The idea of analysing normative systems in terms of deontic concepts has traditionally inspired many logic frameworks [1,12] and tends to overlook the potestative dimension present in normative discourse. This tendency can be also observed in computational systems that incorporate normative expressions, since they generally rely on the idea that (not) being granted *permission* to do

[1] Corresponding author: `g.sileno@uva.nl`.

a certain action upon the system is the same as (not) holding *power* to do it on the system. [2]

Yet, exceptions to this trend in the formal literature exist and are increasing—mainly in the tradition called the theory of normative positions [27] which is based on the characterisation of Hohfeld [13]. For instance, Lindahl [18] interprets power as a *possibility*, which can be either a permission, or a practical possibility, or a legal possibility. Jones and Sergot [14] treat power as a *count-as conditional*: within a given institution, some behaviours by certain parties count as ways of establishing normatively relevant states-of-affairs (e.g. within a department, secretaries' signatures count as their employers' signatures). Markovich [21] provides a definition of power as a *potential* involving an operator for *legal necessity*, which indicates that a party p has power on a party q when a certain behaviour of p brings legal consequences on some normative relation involving q and other parties. Dong and Roy [6,7] emphasize the relation-changing nature of power by defining it in a framework of *dynamic epistemic logic*, where actions available to some agents may affect the normative relations among others. Sileno and Pascucci [30] provide a definition of power in terms of *ability* and, in subsequent work [24], they build diagrams of opposition for various concepts of power (change-centered, outcome-centered and force-centered) and analyse their interactions. Kulicki, Trypuz and Sergot [15] use *labelled transition systems* in order to represent an agent's power to exercise a right in situations where such a right conflicts with those of other parties (e.g. a woman must be able to exercise her right to abortion despite doctors' conscience clause). Similarly, in the technical literature, new policy specification languages have been recently proposed, which includes potestative and deontic concepts, as for instance Symboleo [28] and eFLINT [31]. [3]

All mentioned approaches focus on representing the *way in which power produces changes in normative relations*; yet, they do not address the problem of characterizing the *reasons why power arises in institutional settings*. In other words, we lack a formal theory concerning the "instrumental" nature of power with respect to maintaining normative systems or producing a desired change in them. This is the problem we address in the present article and which, in turn, is related to the problem of appropriately characterizing the difference between the notions of permission ('being allowed') and power ('being enabled').

While permission and power indeed frequently come together, there are institutional settings in which they are activated by different conditions—proving the need for their ontological separation— as observed e.g. by Makinson [19], or Jones and Sergot [14]. For instance, according to canon law, people who are ordained priests retain the sacramental powers even when they leave priesthood: they are merely not allowed to exercise them. This shows that an ac-

[2] See e.g. `GRANT PRIVILEGE` to database users in MySQL, `Permit` effects in XACML rules for access control, `allow` directives in `.htaccess` for Apache webservers.

[3] Insights on how to structure a theory of power independently from a theory of deontic directives in computational settings can be found e.g. in Sileno, Boer and van Engers [29].

tion may produce (normative) effects even in presence of a prohibition that is meant to signal the undesirability of those effects. A similar observation can be drawn also on scenarios involving non-institutional (e.g. physical) actions that are normatively regulated while, obviously, no institutional power is associated to their performance. For instance, the action of smoking is in some contexts permitted although its performance does not involve any previous assignment of power. Another example of the separation between power and permission can be taken from markets operating on digital infrastructures, where enforcement occurs primarily *ex-ante*, in the tradition of authorization systems: transactions are allowed/enabled or not depending on certain conditions. However, markets also open the possibility of fraudulent schemes (i.e. complex behavioural patterns disrupting the normal functioning of the market), whose acknowledgement occurs mostly *ex-post*. Because of the impossibility of strict control (e.g. part of the scheme occurs off-chain), there may be outcomes which are prohibited, and yet possible, as they are in practice enabled by the infrastructure.

In order to analyse how power arises and to which extent the notions of power and permission diverge in normative settings, we introduce a formal framework where a normative system is seen as a collective agent, embodying the institution, that allocates abilities (corresponding to institutional powers) in order to promote the fulfillment of certain desires (corresponding to deontic directives). [4] This simplifying conceptual step is meant to ease the usage of practical reasoning constructs, generally discussed from an individual agent's stanpoint, rather than positing a perfect alignment between intentional and normative categories.

Our framework presents several mechanisms by means of which power originates; these mechanisms are expressed in the form of *conditional rules* and can be grouped into theories. The justification of the mechanisms comes from *reasonability* or *rationality* aspects of the norm-making process, which emerged already in the views of Georg Henrik von Wright [32,33] in terms of the suggested reading of deontic logic: a rational legislator does not create norms saying both $O\phi$ and $O\neg\phi$. Principles of rationality are also at the base of instrumental (or means-end) reasoning, as well as of engineering efforts relying on this, exemplified in control theory and goal-driven agents in AI, upon which we will build for this paper. The proposed formal theories will then have two readings: normative in the sense of the requirements rationality puts on the collective agent, and descriptive as specifying what these "rational" patterns are, implemented accordingly.

Moreover, specifications of power in legal settings generally consists of three dimensions (see e.g. Hart [11, p. 28]): *qualification* (requirements to be ascribed to a role), *performance* (manner and form in which the power is exercised), and *subject-matter* (variety of rights and duties which may be created or modified).

[4] The metaphor of institutions as agents echoes Hobbes' idea of the Leviathan and can be found also in recent technical works, such as Boella and van der Torre [3].

Since we map institutional powers to the abilities of a collective agent to *cause* changes in the normative system (and in particular, changes concerning potential changes), this paper will elaborate on what supports the creation or modification mechanisms associated to performances.

The structure of the article is as follows. Section 2 presents the formal framework, whose main ingredients are: the collective agent's desires, conditions holding in the world and causal connections between events and conditions. Section 3 provides conditional rules that can be used to build a theory for instrumental reasoning and that are grouped into patterns, according to the general mechanisms they represent. Section 4 provides additional patterns of rules corresponding to more specific institutional mechanisms. Finally, Section 5 presents an *answer set programming* (ASP) implementation of the framework.

2 Formal framework

We introduce a formal language $\mathcal{L}$ whose vocabulary and formulas are based on the syntax of the expressions used in ASP. The advantage of this choice is that the framework can be directly encoded into a program, as shown in the final part of the article.

2.1 Vocabulary

Each item in the vocabulary of $\mathcal{L}$ is associated with a *type* clarifying its meaning. Types can be *atomic* or *complex*. A complex type is either a *functional* type or a *union* type. A functional type $\mathtt{t}$ is denoted as $(\mathtt{t}_1, \mathtt{t}_2)$, where $\mathtt{t}_1$ and $\mathtt{t}_2$ are (possibly complex) types which respectively constitute the *input* and the *output* of $\mathtt{t}$. A union type $\mathtt{t}$ is denoted as $\mathtt{t}_1|\mathtt{t}_2$ and can be associated with symbols which are either of type $\mathtt{t}_1$ or of type $\mathtt{t}_2$. The atomic type `boolean` denotes the Boolean truth-values 0 and 1.

Object variables

We use upper case Latin letters for object variables. These variables are associated with two atomic types: `condition` (the type of *conditions* holding in the world, which convey factual information) and `event` (the type of events driven by agents, i.e. actions performed by them). Variables of each type are characterized by a particular notation in the following presentation: Cstand for conditions; A, B stand for events (sometimes with different subscripts making reference to names of individual agents that are part of the normative system, as in A_x, in which case they are actions driven by the individual agent indicated). We take `object` to be the type of all object variables, i.e. the union type `event | condition`. Object variables are denoted as X.

Object constants

We use lower case Latin letters for object constants. They are associated with the two types also used for object variables, namely `condition` and `event`. Notation is used accordingly, namely c for conditions and a for events (sometimes with different subscripts making reference to names of individual agents that are part of the normative system, as in a_x, in which case they are actions driven by the indicated agent).

Connectives

We use unary connectives for classical negation ($\neg$) and default negation ($\mathtt{not}$), binary connectives for classical conjunction ($\wedge$) and the ASP conditional ($\rightarrow$), and the universal quantifier ($\forall$). Unary connectives are associated with the type $(\mathsf{boolean}, \mathsf{boolean})$, binary connectives with the type $((\mathsf{boolean}, \mathsf{boolean}), \mathsf{boolean})$ and, for any object variable X, the expression $\forall X$ is associated with the type $(\mathsf{boolean}, \mathsf{boolean})$. [5] In the present context, the interpretation of $\rightarrow$ can be either *descriptive*, if one wants to characterize how an idealized normative system works, or *prescriptive*, if one wants to characterize how a normative system should be designed.

Function symbols

We use two binary function symbols $causes^+$ and $causes^-$ which take as input an event (first argument) and a condition (second argument). Their outputs are conditions. Thus, their type is: $((\mathsf{event}, \mathsf{condition}), \mathsf{condition})$. The function $causes^+$ represents a positive form of causation: an expression of the form $causes^+(A, C)$ reifies a causal mechanism binding action type A and condition C. More precisely, it indicates a condition according to which, by performing an action of type A, the agent triggers the consequent realization of C. The function $causes^-$ represents a negative form of causation: the expression $causes^-(A, C)$ reifies an inhibiting causal mechanism binding action type A and condition C. It indicates a condition according to which, by performing an action of type A, the agent inhibits the consequent realization of C (by any other means). We also use a unary function symbol neg which takes a condition as input and gives a condition (incompatible with C) as output. [6] Hence, its type is: $(\mathsf{condition}, \mathsf{condition})$.

Predicate symbols

Our framework involves reference to desires with a positive or negative attitude. We introduce two predicates Des^+ and Des^- which take a condition or an event as input and give a truth-value as output. Hence, their type is: $(\mathsf{object}, \mathsf{boolean})$. $Des^+(C)$ means that the collective agent has a positive attitude towards condition C (e.g. it prefers C to hold), whereas $Des^-(C)$ means that the collective agent has a negative attitude towards condition C (e.g. it prefers C to not hold). Analogous readings hold for $Des^+(A)$ and $Des^-(A)$. Moreover, we use a unary predicate $Holds$ taking a condition as input and giving a truth-value as output. Hence, its type is: $(\mathsf{condition}, \mathsf{boolean})$. The meaning of an expression of the form $Holds(C)$ is that condition C holds.

---

[5] We stress that the connective $\rightarrow$ denotes a conditional operator typically used in ASP programs and behaving differently from material implication; for details, see [17]. Material implication (as well as the other classical connectives and $\exists$) is definable in $\mathcal{L}$ via the primitive connectives.

[6] The derivation mechanism (e.g. in our ASP implementation) may rely on intensional predicate functions, and therefore may not require to determine this incompatible condition.

Finally, we use a symbol for identity among objects (=). The type of = is ((`object`, `object`), `boolean`).

2.2 Terms, formulas and instrumental theories

Definition 2.1 *Terms.* The set of terms (T_1, T_2, etc.) of $\mathcal{L}$ is the smallest set satisfying the following properties:

- every object variable and object constant is a term of $\mathcal{L}$;
- if T is a term whose type is condition, then $neg(T)$ is a term of $\mathcal{L}$; [7]
- for every term T_1 of type event and term T_2 of type condition, $causes^+(T_1, T_2)$ and $causes^-(T_1, T_2)$ are terms of $\mathcal{L}$.

Definition 2.2 *Formulas.* The set of atomic formulas of $\mathcal{L}$ is the smallest set satisfying the following properties:

- for every terms T_1 and T_2, $T_1 = T_2$ is a formula of $\mathcal{L}$;
- for every term T, $Des^+(T)$ and $Des^-(T)$ are formulas of $\mathcal{L}$;
- for every term T whose type is condition, $Holds(T)$ is a formula of $\mathcal{L}$;
- if ϕ is a formula of $\mathcal{L}$, then so are $\neg\phi$ and $\texttt{not}(\phi)$;
- if ϕ and ψ are formulas of $\mathcal{L}$, then so are $\phi \wedge \psi$ and $\phi \rightarrow \psi$;
- if ϕ is a formula of $\mathcal{L}$, then so is $\forall X : \phi$, for X an object variable.

A formula is atomic iff it has one of the forms $T_1 = T_2$, $Des^+(T)$, $Des^-(T)$ or $Holds(T)$. We use $T_1 \neq T_2$ as an abbreviation for $\neg(T_1 = T_2)$. In the construction of terms and formulas the auxiliary symbols ':', '(' and ')' can be omitted according to binding conventions in the ASP syntax [17].

Definition 2.3 *Instrumental theories.* An instrumental theory Θ is a non-empty set of formulas which are either atomic or of the form $\phi \rightarrow \psi$ and, in the latter case, ψ is either of the form $Des^+(T)$ or of the form $Des^-(T)$. Within a theory, a formula of the form $\phi \rightarrow \psi$ is said to be a *conditional rule* and ψ is said to be the *target desire* of that rule. If Θ consists only of atomic formulas, then it is said to be an *atom-based* theory. Finally, a theory Θ_1 is an *expansion* of a theory Θ_2 iff $\Theta_1 \supseteq \Theta_2$.

Let Θ be an atom-based theory, namely a set of formulas describing either the collective agent's desires, or conditions holding in the world, or causal connections between events and conditions, or identity of objects. It is possible to expand Θ to a theory Θ' by adding conditional rules representing relevant mechanisms for instrumental reasoning that can be grouped into certain patterns. The next section will explain how this expansion can be performed. In all rules mentioned therein, free occurrences of variables should be understood as being in the scope of a universal quantifier, as in ASP rules [17].

---

[7] A technical remark: in the ASP implementation of the framework (Section 5) will only make use of terms where *neg* occurs at most once, since this is enough to encode the rules discussed in the article.

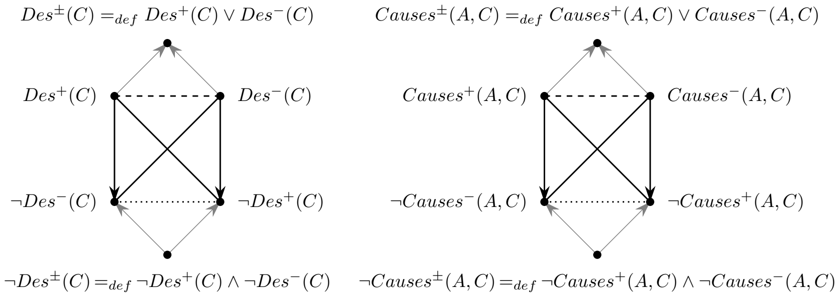

Fig. 1. Relations between elements of the language illustrated on hexagons of opposition. Labels of the hexagon on the right abuse the notation not to overload the image: e.g. $Causes^{+}(A, C)$ has to be read as $Holds(causes^{+}(A, C))$. Given two vertices v and u, an arrow from v to u indicates that u is a subalternate of v; a full line between v and u that v and u are contradictories; a dashed line between v and u that v and u are contraries; a dotted line between v and u that v and u are sub-contraries. The contrariety relation on the left hexagon holds only for collective agents corresponding to idealized normative systems; yet, we will also discuss scenarios in which a collective agent happens to have conflicting desires. The logic assumed for desires and causation is very simple and sufficient to serve the purposes of the article.

3 Patterns of conditional rules and theory expansion

3.1 Fundamental patterns

Suppose that the collective agent has certain desires with either positive or negative attitude with respect to a certain outcome (represented by condition C), and suppose that this condition currently is not in place. Moreover, suppose that action A is a causal mechanism producing C or inhibiting its production. Thus, we will start with an atom-based theory Θ that will contain one among $Des^{+}(C)$ and $Des^{-}(C)$, as well as one among $Holds(causes^{+}(A, C))$ and $Holds(causes^{-}(A, C))$, as well as $\neg Holds(C)$. The following patterns show how the collective agent's additional desires are triggered by the various combinations of these options. We stress an important point on the interpretation of conditional rules mentioned in the article. Under the descriptive reading of operator $\rightarrow$, a rule states that the target desire arises for the collective agent as an indication of a *sufficient* (rather than a *necessary*) instrument to get the desired outcome. Similarly, under the prescriptive reading of operator $\rightarrow$, a rule states that a *prima facie* (rather than an *all-things-considered*) duty arises for the collective agent with respect to the target desire. The reason behind this is that a sufficient instrument (a *prima facie* duty) does not have to be necessarily used (fulfilled) to achieve the intended goal, given that sometimes such an instrument (duty) might lead to unwelcome outcomes too. [8]

---

[8] We thank one of the reviewers for DEON for inquiring about this aspect of our framework.

(i) Means-end derivation

If you desire some (not present) condition to hold, and you have the ability to make this happen, you [should] desire to use such ability: [9]

$$\textbf{[1]} \quad Des^{+}(C) \wedge \neg Holds(C) \wedge Holds(causes^{+}(A, C)) \rightarrow Des^{+}(A)$$

The formula in the antecedent of the conditional is a conjunction composed of three elements that represent a common template in control theory and classic AI [10]: the *reference* (i.e. $Des^{+}(C)$) and the *current state* (i.e. $\neg Holds(C)$) define the trajectory, the *causal mechanism* (i.e. $Holds(causes^{+}(A, C))$) identifies the control that produces that trajectory. The consequent of the conditional indicates that the causal mechanism needs to be triggered. [11] Following a similar rationale, three additional rules can be identified:

$$\textbf{[2]} \quad Des^{-}(C) \wedge \neg Holds(C) \wedge Holds(causes^{+}(A, C)) \rightarrow Des^{-}(A)$$
$$\textbf{[3]} \quad Des^{-}(C) \wedge \neg Holds(C) \wedge Holds(causes^{-}(A, C)) \rightarrow Des^{+}(A)$$
$$\textbf{[4]} \quad Des^{+}(C) \wedge \neg Holds(C) \wedge Holds(causes^{-}(A, C)) \rightarrow Des^{-}(A)$$

Rule **2** is about the derivation of negative desires: actions which may trigger undesired events are undesired too. Rule **3** and Rule **4** deal with inhibiting mechanisms ($causes^{-}$), which are desired if they disable the occurrence of negatively desired events, undesired otherwise.

Next, we consider rules to represent situations in which the collective agent desires an instrument to obtain a certain goal.

(ii) Desire of instrument

If you desire some (not present) condition to hold, but you do not have the ability to make it happen, you [should] desire to create this ability. From a conceptual point of view, this rule works at a meta-level with respect to the previous ones; it indicates that before having the possibility of using a means to reach an end, we should have some means available. Similarly, *if you desire some (not present) condition to hold, and you have the ability to make it happen, you [should] desire not to remove this ability.* This second rule indicates

[9] The *should* in the various patterns indicates the normative (as opposed to descriptive) possible readings.

[10] See e.g. the general template of negative feedback [8, p. 8], the general architecture of goal-based agents [25, p. 56], or agent-programming language frameworks [4].

[11] Computational implementations based on practical reasoning principles typically set an *intentional bottleneck* constraint, because the agent triggers only one action to reach the desired outcome (e.g. the best one, in terms of costs and certainty). This passage, from possibly conflicting volitional elements (desires) to non-conflicting deliberative elements (intentions), is well-known in *beliefs-desires-intentions* (BDI) frameworks [5], and has strong connections to argumentative patterns (*necessary* vs *sufficient means*) [10,34,16], and to the distinction between *prima-facie* and all-things-considered obligations [2], of which we talked at the beginning of this section with respect to the two alternative readings. For simplicity, we keep these aspects out of our current scopes, although we acknowledge their relevance and plan to approach them in future works.

instead that when such a means is available, we should protect it. However, these patterns do not say anything about how the relevant causal mechanisms are created, and unnecessarily complicates the definition of $\mathcal{L}$. For this reason, we will rather focus on the following *realization* pattern in order to expand the theory Θ with mechanisms to create instruments for desired goals.

(iii) Creation of instrument

If you desire some (not present) condition to hold, and you do not have the ability to make it happen, but you have the ability to create such an ability, you [should] desire to use this ability.

$$\begin{aligned} Des^+(C) \wedge \neg Holds(C) \wedge \texttt{not}\ \exists A : Holds(causes^+(A,C)) \wedge \\ Holds(causes^+(B, causes^+(A,C))) \rightarrow Des^+(B) \end{aligned}$$

This pattern looks at the existing causal mechanisms and picks some action available to produce the target causal mechanism (enabling, or disabling a change).[12] This schema could in principle be applied recursively to take into account higher-order levels (e.g. causal mechanisms that create causal mechanisms that create...), but this is out of the scope for our present purposes.

The rules resulting from all combinations of desires, conditions and causal relations are the following:

$$\begin{array}{rl}
\text{(for 5–8)} & \textit{let}\ \phi\ \textit{be}\ \neg Holds(C) \wedge \texttt{not}\ \exists A : Holds(causes^+(A,C)),\ \textit{then} \\
\textbf{[5]} & \phi \wedge Des^+(C) \wedge Holds(causes^+(B, causes^+(A,C)))) \rightarrow Des^+(B) \\
\textbf{[6]} & \phi \wedge Des^+(C) \wedge Holds(causes^-(B, causes^+(A,C)))) \rightarrow Des^-(B) \\
\textbf{[7]} & \phi \wedge Des^-(C) \wedge Holds(causes^+(B, causes^+(A,C)))) \rightarrow Des^-(B) \\
\textbf{[8]} & \phi \wedge Des^-(C) \wedge Holds(causes^-(B, causes^+(A,C)))) \rightarrow Des^+(B) \\
\text{(for 9–12)} & \textit{let}\ \psi\ \textit{be}\ \ \neg Holds(C) \wedge \texttt{not}\ \exists A : Holds(causes^-(A,C)),\ \textit{then} \\
\textbf{[9]} & \psi \wedge Des^+(C) \wedge Holds(causes^+(B, causes^-(A,C)))) \rightarrow Des^-(B) \\
\textbf{[10]} & \psi \wedge Des^+(C) \wedge Holds(causes^-(B, causes^-(A,C)))) \rightarrow Des^+(B) \\
\textbf{[11]} & \psi \wedge Des^-(C) \wedge Holds(causes^+(B, causes^-(A,C))) \rightarrow Des^+(B) \\
\textbf{[12]} & \psi \wedge Des^-(C) \wedge Holds(causes^-(B, causes^-(A,C))) \rightarrow Des^-(B)
\end{array}$$

Rules **5-8** deal with the absence of causal mechanisms bringing about change; Rules **9-12** deal with the absence of inhibiting mechanisms.

(iv) Protection of instrument

We apply the same principle with the second pattern in (ii), concerning the protection of an existing relevant ability. We obtain therefore eight additional

[12] Note the use of connective `not` in this formula, which indicates that no such instrument has been found.

rules:

(for 13–16) *let* ϕ *be* $\neg Holds(C) \wedge Holds(causes^{+}(A,C))$, *then*

[13] $\phi \wedge Des^{+}(C) \wedge Holds(causes^{+}(B, neg(causes^{+}(A,C)))) \rightarrow Des^{-}(B)$

[14] $\phi \wedge Des^{+}(C) \wedge Holds(causes^{-}(B, neg(causes^{+}(A,C)))) \rightarrow Des^{+}(B)$

[15] $\phi \wedge Des^{-}(C) \wedge Holds(causes^{+}(B, neg(causes^{+}(A,C)))) \rightarrow Des^{+}(B)$

[16] $\phi \wedge Des^{-}(C) \wedge Holds(causes^{-}(B, neg(causes^{+}(A,C)))) \rightarrow Des^{-}(B)$

(for 17–20) *let* ψ *be* $\neg Holds(C) \wedge Holds(causes^{-}(A,C))$, *then*

[17] $\psi \wedge Des^{+}(C) \wedge Holds(causes^{+}(B, neg(causes^{-}(A,C))) \rightarrow Des^{+}(B)$

[18] $\psi \wedge Des^{+}(C) \wedge Holds(causes^{-}(B, neg(causes^{-}(A,C)))) \rightarrow Des^{-}(B)$

[19] $\psi \wedge Des^{-}(C) \wedge Holds(causes^{+}(B, neg(causes^{-}(A,C)))) \rightarrow Des^{-}(B)$

[20] $\psi \wedge Des^{-}(C) \wedge Holds(causes^{-}(B, neg(causes^{-}(A,C)))) \rightarrow Des^{+}(B)$

Patterns similar to (i), (iii) and (iv) can be constructed also *in presence* of the target condition C i.e. when the antecedent of a conditional rule includes $Holds(C)$. By making this amendment, one obtains twenty additional rules **[21–40]** (more precisely, for $\mathbf{1 \leq i \leq 20}$, rule $\mathbf{20 + i}$ is obtained by performing the mentioned amendment on rule **i**). These new rules are here not explicitly stated for reasons of space, but are present in the code. For more details, see Table 1.

3.2 Relaxations

All the patterns identified above require a desire to be present, either positively, either negatively, in order to trigger the derivation of new desires. Even if at the moment the agent does not desire something (the negation here stands as absence of desire), it is not the case that the agent has a negative volitional attitude towards that thing. This is a more relaxed condition, as indicated by the hexagon of opposition for desires (Fig. 1): $\neg Des^{-}(C)$ subalternates $Des^{+}(C)$. This relation identifies $\neg Des^{-}(C)$ as a *a necessary condition for* $Des^{+}(C)$ *to hold.* We may therefore explore the possibility of expanding theory Θ with new patterns involving the relaxed condition $\neg Des^{-}(C)$. We will distinguish two families of scenarios: contextual and direct anticipatory interventions.

3.2.1 Contextual interventions: preparing relevant abilities

In the first family of scenarios, the *anticipation* approach can be read as such: if the agent does not have a negative volitional attitude towards C, it is expected that at a certain point $Des^{+}(C)$ may hold, and so better be prepared by settling what will then be needed to bring about C. In other words, to motivate such endeavour it is sufficient to be committed to $\neg Des^{-}(C)$. In this way, once the desire comes to existence, relevant abilities are already in place for the agent. For instance, relaxing pattern **[5]** we obtain:

[5*] $\neg Des^{-}(C) \wedge \neg Holds(C) \wedge \texttt{not}\ \exists A : Holds(causes^{+}(A,C)) \wedge Holds(causes^{+}(B, causes^{+}(A,C))) \rightarrow Des^{+}(B)$

Following the same idea we can rewrite all patterns [**5**–**20**], [**25**–**40**] into relaxed forms [**5***–**20***], [**25***–**40***], which are about enabling or disabling (positive or negative) causal mechanisms that will be relevant when C—acknowledged as *desirable* (positively or negatively)—is eventually instantiated.

3.2.2 Direct interventions

The second family of scenarios concerns the core of the means-end derivation. Relaxing as before the premises concerning desire in e.g. Rule [**1**] and Rule [**2**], we obtain:

$$[\mathbf{1*}] \quad \neg Des^{-}(C) \wedge \neg Holds(C) \wedge Holds(causes^{+}(A, C)) \rightarrow Des^{+}(A)$$
$$[\mathbf{2*}] \quad \neg Des^{+}(C) \wedge \neg Holds(C) \wedge Holds(causes^{+}(A, C)) \rightarrow Des^{-}(A)$$

At further inspection, however, we observe that patterns concerning action avoidance (as [**2***]) have different practical implications than patterns concerning performance of actions ([**1***]). For instance, if C is recognized as *undesirable* (but possibly not as *undesired*), it is already acceptable that the agent should avoid performing an action A that would bring about C.[13] In contrast, if C is recognized as *desirable* (but possibly not as *desired*), it is less sound that the agent should perform an action to already bring about the condition C, just as we were in the positive desire case. A possible explanation for this intuition is that, independently of its effects in the world, executing an action carries always costs for the agent, whereas action avoidance generally plays a role only in plan selection, not in execution. In other words, the preference about applying the proposed relaxation between the two patterns seems to emerge out of principles of economy, entailing that instrumental reasoning common-sensically carries along a concurrent desire to select efficient plans (all other things being the same). This trail of thoughts is confirmed by observing that the avoidance case becomes unsound just as the performance case when there are few or no other plans available (or those that are available have much higher cost than the avoided plan).

Focusing for simplicity only on the qualitative dimension of these conditions, we can capture them as absence or presence of a *substitute* (equivalent and alternative) ability:

$$Holds(onecauses^{+}(A, C)) =_{def} Holds(causes^{+}(A, C)) \wedge \texttt{not}\ \exists B : Holds(causes^{+}(B, C)) \wedge A \neq B$$
$$Holds(manycauses^{+}(A, C)) =_{def} Holds(causes^{+}(A, C)) \wedge \exists B : Holds(causes^{+}(B, C)) \wedge A \neq B$$

The patterns can then be rewritten as:

$$[\mathbf{1**}] \quad \neg Des^{-}(C) \wedge \neg Holds(C) \wedge Holds(onecauses^{+}(A, C)) \rightarrow Des^{+}(A)$$
$$[\mathbf{2**}] \quad \neg Des^{+}(C) \wedge \neg Holds(C) \wedge Holds(manycauses^{+}(A, C)) \rightarrow Des^{-}(A)$$

[13] Note that we are overlooking all problems related to defeasibility here. In the general case, the action A may still be required for satisfying concurrent desires of higher priority.

		Absence of target: $\neg Holds(C)$		Presence of target: $Holds(C)$	
	Available ability	$Des^+(C)$ or $\neg Des^-(C)$	$Des^-(C)$ or $\neg Des^+(C)$	$Des^+(C)$ or $\neg Des^-(C)$	$Des^-(C)$ or $\neg Des^+(C)$
`k.`	$causes^-(A, neg(C))$			$Des^+(A)$	$Des^-(A)$
`r.`	$causes^+(A, C)$	$Des^+(A)$	$Des^-(A)$		
`e.`	$causes^+(A, neg(C))$			$Des^-(A)$	$Des^+(A)$
`a.`	$causes^-(A, C)$	$Des^-(A)$	$Des^+(A)$		
`kk.`	$causes^-(B, neg(causes^-(A, neg(C))))$			$Des^+(B)$	$Des^-(B)$
`kr.`	$causes^-(B, neg(causes^+(A, C)))$	$Des^+(B)$	$Des^-(B)$		
`ke.`	$causes^-(B, neg(causes^+(A, neg(C))))$			$Des^-(B)$	$Des^+(B)$
`ka.`	$causes^-(B, neg(causes^-(A, C)))$	$Des^-(B)$	$Des^+(B)$		
`rk.`	$causes^+(B, causes^-(A, neg(C))$			$Des^+(B)$	$Des^-(B)$
`rr.`	$causes^+(B, causes^+(A, C))$	$Des^+(B)$	$Des^-(B)$		
`re.`	$causes^+(B, causes^+(A, neg(C))$			$Des^-(B)$	$Des^+(B)$
`ra.`	$causes^+(B, causes^-(A, C))$	$Des^-(B)$	$Des^+(B)$		
`ek.`	$causes^+(B, neg(causes^-(A, neg(C))))$			$Des^-(B)$	$Des^+(B)$
`er.`	$causes^+(B, neg(causes^+(A, C)))$	$Des^-(B)$	$Des^+(B)$		
`ee.`	$causes^+(B, neg(causes^+(A, neg(C))))$			$Des^+(B)$	$Des^-(B)$
`ea.`	$causes^+(B, neg(causes^-(A, C)))$	$Des^+(B)$	$Des^-(B)$		
`ak.`	$causes^-(B, causes^-(A, neg(C)))$			$Des^-(B)$	$Des^+(B)$
`ar.`	$causes^-(B, causes^+(A, C))$	$Des^-(B)$	$Des^+(B)$		
`ae.`	$causes^-(B, causes^+(A, neg(C)))$			$Des^+(B)$	$Des^-(B)$
`aa.`	$causes^-(B, causes^-(A, C))$	$Des^+(B)$	$Des^-(B)$		

Table 1
Overview of all instrumental patterns identified in section 2, in presence/absence of desired/undesired target, and considering presence/absence of four types of relevant abilities (`k`, `r`, `e`, `a`, standing respectively for *keep, reach, escape, avoid*). The relaxations from $Des^+(C)$ to $\neg Des^-(C)$ and from $Des^-(C)$ to $\neg Des^+(C)$ requires conditions on absence and on presence of substitute abilities (see section 2.2).

In words, performance is sustained by the absence of equivalent alternatives (although for an individual agent this desire is typically defeated due to economic reasons); action avoidance is sustained by the presence of equivalent alternatives (and generally it is not defeated as it does not incur in further costs). This idea will be applied on patterns **[1–4]** and **[21–24]**.

3.3 A framework of interventions

The 40 patterns can then be reorganized as follows. We first separate scenarios in which the target condition C is present from those in which it is absent, then we specify the attitude towards the target (positive or negative). The resulting organization is illustrated in Table 1. The evident symmetries suggest that further simplification of the notation is possible at a syntactic level, but this is beyond the scope of the present paper.

The practical derivation performed by the agent eventually depends on the available abilities, reified as causal connections, or possible interventions. For better readability, these abilities can labeled: `k` for *Keep* abilities, as e.g. $causes^-(A, neg(C))$, maintaining C in $Holds(C)$; `r` for *Reach* abilities, as e.g. $causes^+(A, C)$, producing C in $\neg Holds(C)$; `e` for *Escape* abilities, as e.g. $causes^+(A, neg(C))$, removing C in $Holds(C)$; `a` for *Avoid* abilities, as e.g.

$causes^{-}(A, C)$, inhibiting C in $\neg Holds(C)$.[14] This relabeling is useful here to denote in a more succint manner second-order abilities. For instance, `rr` will denote the ability to "reach" an ability to "reach" a certain condition—that is, $causes^{+}(B, causes^{+}(A, C))$.

4 Relevant institutional patterns

This section elaborates on how the machinery presented above can be applied to investigate patterns observable in institutional domains.

4.1 From (collective) agent to normative system

At this point, we want to interpret agentive attitudes in terms of a normative system, here taken as (i) a system of agents guided by (ii) a system of norms. Intuitively, the collective agent's desires would map to deontic directives, its abilities to potestative directives. Yet, two considerations are crucial in this passage. First, we need to take into account that there exist abilities which are *primitives* or given independently from the institution: either because they are physical abilities proper of individuals, or because they are (recognized) institutional abilities provided by some other institution. Second, actions A performed by the collective agent map to actions performed by individual agents *for the sake of the institution*. However, because they are autonomous, individuals may still perform actions for other purposes (e.g. for their own interests). We will utilize subscripts to distinguish individuals, e.g. A_x would be an event driven by agent x. We will now consider a few relevant patterns to show potential applications of the proposed framework.

4.2 Protected liberty

In the normative system literature, a distinction is usually made between permissions (and/or liberties) which are explicitly declared, and those which are derived from the absence of relevant obligations or prohibitions. Various authors[15] have argued that a *permissive norm* issuing a "strong permission" bring along additionally mechanisms, that can be overall reorganized as:

---

[14] This framework is similar to taxonomies presented in other disciplines. For instance, works in agent-based programming distinguish *maintenance*, *achievement*, *remedy*, and *avoidance* goals. Similarly, in psychology, Ogilvie and Rose [23] introduce the *prevent-acquire-cure-keep* (PACK) framework to classify explanations given by people about their own behaviour. The PACK framework of motives however takes also into account the positive or negative attitude of the agent towards the target. For instance *acquire* (A) is always about reaching a positive outcome; the agent is not deemed to reach (purposely) a negative outcome.

[15] For instance, Makinson [19] observes that explicit permission "appears to be needed by real-life normative systems that change over time, as a device for limiting the interpretation of obligations and preventing their proliferation." Together with Alchourrón and Bulygin, he sees this practice as "to limit the authority of subordinate instances to create new norms" that go against the given permission. Additional mechanisms are summarized by Sartor [26], building upon Hart (for the protection coming with explicit permissions), Alexy and Pettit (for protected freedoms), and Sen (for the creation of effective capability in presence of permission). More recently, Markovich and Roy provide a logical formalization of the freedom of thought pointing out all the protective layers [22].

- a *practical protection*: the prohibition of interference against performance addressing all other social participants;
- an *institutional protection*: the disability for subordinate regulators to produce directives conflicting with that norms;

A special case is then that of institutional actions; here explicit liberty (e.g. right to marry discussed by Markovich [20]) also implies an obligation of the normative system to follow along the consequence of the action, i.e.

- an *institutional instrument*, i.e. a power for the addressee of the liberty to require a certain performance from the normative system.

A non-intervention by the normative system (the only one that can keep track of effects of institutional actions as marrying) would count as an interference.

For instance, let us consider a directive as *x is free to marry, as well as free not to marry.* The practical protection function entails that someone should not prohibit or interfere with x's marrying, as well as nobody should oblige or control x in this sense. The institutional protection entails that subordinate regulators cannot change this directive. The institutional instrument entails that x should be enabled to marry if x wished to (the institution being the only agency able to produce this institutional outcome).

4.2.1 Application of the proposed framework

Rephrasing this discussion in agentive terms, a weak liberty would map to having $\neg Des^{\pm}$ derived from the absence of other volitional attitudes (i.e. issued by some form of default negation), whereas a strong liberty would map to stating the attitude $\neg Des^{\pm}$ (entailing strongly negated statements). The directive expressed in the example is an explicit expression of strong liberty, in the form of a $\neg Des^{\pm}(C)$ position:

$$\neg Des^{\pm}(C) \rightarrow \neg Des^{+}(C) \wedge \neg Des^{-}(C)$$

Looking at Table 1, this entails that both conclusions in the first and the second column are potentially relevant. The role of the constraints on relaxation (section 2.2) based on substitute abilities becomes here particularly relevant, as they prevent to conclude opposite desires.

As a first validation, let us check whether our framework captures what is expected from the legal literature. With respect to *practical protection*, we have:

- If an event A_y can inhibit the outcome C (e.g. y can interfere with x's marrying), knowing that A_x also can do it (x can refrain from marrying), entails that A_y is undesired (Table 1, `a`, first column).
- If an event A_y can bring about the outcome C (e.g. y has the ability to control marrying besides x), then A_y is undesired (Table 1, `r`, second column).

With respect to *institutional protection*:

- The institution needs to protect x's ability by disabling the possibility to remove it (Table 1, `kr`, first column).

Note that in this case there are no substitute abilities (only the institution can intervene on this matter) and this entails that the conclusion is a positive desire. With respect to *institutional instrument*,

- The institution needs to put in place mechanisms so that x's marrying is eventually acknowledged (Table 1, `rr`, first column)

Note that the framework concludes also that the institution should abstain to create or keep instruments that may interfere with the marriage (Table 1, `ka` and `ra`, first column), as long as x is expected to have the ability of abstaining from marrying. Dually, our framework would suggest that if marriages may be combined (and individuals have no ability to abstain from marrying), dedicated institutional instruments should be created to empower individuals to stop marriages to occur. The legal theoretical understanding of such a conclusion could be of course subject to a detailed discussion, but we only refer to the openness of this question here.

5 Implementation

We have implemented a version of the framework in *answer set programming* (ASP) [17].[16] We will report here excerpts of the code, the full version is publicly available.[17] The notation used in the code is slightly adapted from the formal framework, following syntactic conventions of the ASP syntax (`-` represents $\neg$, `posdes` represents Des^+, `poscauses` represents $causes^+$, `nodes` represents $\neg Des^\pm$, all predicates start with lower case letters, `:-` replaces $\rightarrow$ switching antecedent and consequent, etc.). For instance, rule **[1]** becomes:

```
posdes(A) :- posdes(C), -holds(C), holds(poscauses(A, C)). % 1 (r)
```

Stating that there is no available cause of a condition requires using *default negation* (`not`), as in the second line of the code below:

```
holds(some_poscauses(C)) :- holds(poscauses(_, C)).
posdes(B) :- posdes(C), -holds(C), not holds(some_poscauses(C)),
             holds(poscauses(B, poscauses(A, C))). % 5 (rr)
```

Negated conditions require a function `neg` operating at the level of terms, e.g.:

```
negdes(A) :- posdes(C), holds(C),
             holds(poscauses(A, neg(C))). % 21 (e)
negdes(B) :- posdes(C), -holds(C), holds(poscauses(A, C)),
             holds(poscauses(B, neg(poscauses(A, C)))). % 13 (er)
```

In order to take into account anticipatory patterns, following the relaxation discussed in Section 3, we first need to define relations corresponding to subalternation in the deontic hexagon of Figure 1:

```
-negdes(C) :- posdes(C). -posdes(C) :- negdes(C).
```

[16] ASP is a declarative programming paradigm based on a *stable-model* semantics [9], oriented towards NP-hard search problems, and increasingly used to model and solve problems in research and industry in a wide range of application domains.

[17] `https://github.com/gsileno/abilities-desires-asp`

```
-negdes(C) :- nodes(C). -posdes(C) :- nodes(C).
```

The two constraints we require for the relaxation are encoded as:

```
holds(one_poscauses(A, C)) :-
   holds(poscauses(A, C)), not holds(many_poscauses(A, C)).
holds(many_poscauses(A, C)) :-
   holds(poscauses(A, C)), holds(poscauses(B, C)), A != B.
```

We can then rewrite the patterns in the relaxed form:

```
posdes(A) :- -negdes(C), -holds(C),
   holds(one_poscauses(A, C)). % 1** (r)
negdes(A) :- -posdes(C), -holds(C),
   holds(many_poscauses(A, C)). % 2** (r)
```

For the combinatorial exploration, we specify that all conditions present in the program may hold or not:

```
{holds(C)} :- condition(C).
```

We also introduce four integrity constraints, one for each type of causal connection. For instance, the potential transition reified in $causes^{+}(A, C)$ requires C not to hold (otherwise there would be no change):

```
-holds(C) :- holds(poscauses(A, C)), condition(C).
```

To reason about the absence of conditions, we need to introduce a closed-world assumption relying on default negation:

```
-holds(C) :- not holds(C), condition(C).
```

With these rules, we can specify a certain normative configuration (deontic and potestative directives mapped to desires and abilities) and automatically derive what theoretically entailed by the instrumental reasoning patterns identified above. The result may confirm and possibly extend normative constructs discussed in the literature. For instance, supposing that x's marrying (event denoted as `m_x`) is permitted (if interpreted as facultativeness, in our framework it is encoded as `nodes(m_x)`), we can check the associated normative consequences in different potestative configurations. For instance, suppose that the authority a has the institutional power to create the power for x to marry, i.e.:

```
nodes(m_x). holds(poscauses(a_a, poscauses(a_x, m_x))).
```

Indeed, we derive that the authority should exercise its power, i.e. `posdes(a_a)`. Now let us assume that this power already exists:

```
nodes(m_x). holds(poscauses(a_x, m_x)). holds(poscauses(a_y, m_x)).
```

If x and y have both the ability to bring about x's marriage (for instance forcing x to do so), y is forbidden to do so, i.e. we derive `negdes(a_y)`. However, this scenario shows also a limitation of the current formalization: as it does not allow for distinguishing individuals driving actions (e.g. legitimate from illegitimate), it generates also a prohibition upon x. We leave this extension to future work.

6 Conclusion

The paper introduces a framework to investigate the relationships between potestative and deontic categories, looking at institutions as collective agents and exploiting instrumental reasoning patterns. The framework allows for performing a combinatorial exploration of several patterns of interactions between powers, obligations, prohibition and permissions. As an example, by performing the derivation on a strong permission/liberty, we entailed several mechanisms discussed (often separately, and by distinct authors) in the normative systems' literature. Yet, we acknowledge that these results are just initial with respect to the potential applications of the framework. At the moment, our analysis is a-temporal. We primarily identify desires "rationally" holding at a certain moment of time, including those concerning the modification of abilities, without being concerned of solving conflicts that may emerge. In future work, we aim to add to the present framework deliberative and causal/temporal modules.

The deliberative module will serve to select a set of (non-conflicting) intentions based upon the existing (possibly conflicting) desires. This problem has connections with the distinction between *prima-facie* vs actual obligations. Indeed, a normative system does not consists only of mechanisms for allocating powers, but also of substantial and procedural constraints on such allocation. In our current formalization, these constraints may be captured as directives which, in the moment of allocation/derivation, would determine conflicts. The causal/temporal module will enable reasoning about the effects of events (including performances driven by intention). Once these modules are integrated with the present framework, we could reason on the overall institutional dynamics. For instance, the need for institutional protection will be derived automatically after an institutional instrument has been decided and created.

Complementary to these extensions, further effort is needed to identify how to unveil institutionally relevant generic mechanisms, as for instance the power of declaring the occurrence of a violation, and of requiring interventions from an enforcer. More fundamentally, the prescriptive reading opens up also to the use of the framework for assistive technologies: given a certain normative system, how could this be improved?

Acknowledgments Giovanni Sileno was partly funded by the Dutch Research Council (NWO) for the HUMAINER AI project (KIVI.2019.006). Matteo Pascucci was supported by the Štefan Schwarz Fund (2020-2023) for the project "A fine-grained analysis of Hohfeldian concepts" and by VEGA grant n. 2/0125/22 "Responsibility and modal logic". Réka Markovich was supported by the Fonds National de la Recherche Luxembourg through the project INTEGRAUTO (INTER/AUDACE/21/16695098).

Authors' contribution The ideas at the basis of this work rely on extended discussion among the three authors. As far as the article production is concerned, Sections 3, 4, 5 and 6 are mainly due to Giovanni Sileno, Section 2 is mainly due to Matteo Pascucci and Section 1 is equally due to all the authors.

References

[1] Åqvist, L., *Deontic logic*, in: D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, 2002 pp. 147–264.

[2] Asher, N. and D. Bonevac, *"Prima facie" obligation*, Studia Logica **57**, pp. 19–45.

[3] Boella, G. and L. van der Torre, *Constitutive norms in the design of normative multiagent systems*, in: F. Toni and P. Torroni, editors, *Computational Logic in Multi-Agent Systems* (2006), pp. 303–319.

[4] Bordini, R., J. Hübner and M. Wooldridge, "Programming Multi-Agent Systems in AgentSpeak using Jason," Wiley Series in Agent Technology, Wiley, 2007.

[5] Bratman, M., "Intention, plans, and practical reason," CSLI publications, 1987.

[6] Dong, H. and O. Roy, *Dynamic logic of power and immunity*, in: J. Baltag, A. Seligman and T. Yamada, editors, *Logic, Rationality, and Interaction*, 2017 pp. 123–136.

[7] Dong, H. and O. Roy, *Dynamic logic of legal competences*, Journal of Logic, Language and Information **30** (2021), pp. 701–724.

[8] Doyle, J. C., B. A. Francis and A. R. Tannenbaum, "Feedback control theory," Macmillan Publishing Co., 1990.

[9] Gelfond, M. and V. Lifschitz, *The stable model semantics for logic programming*, Proceedings of International Logic Programming Conference and Symposium (1988), pp. 1070–1080.

[10] Hare, R. M., "Practical inferences," MacMillan Press Ltd, 1971.

[11] Hart, H. L. A., "The concept of law," Oxford University Press, 1994, second edition.

[12] Hilpinen, R. and P. McNamara, *Deontic logic: a historical survey and introduction*, in: D. Gabbay, J. Horty, X. Parent, R. van der Meyden and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, 2013 pp. 1–134.

[13] Hohfeld, W. N., *Fundamental legal conceptions as applied in judicial reasoning*, The Yale Law Journal **26** (1917), pp. 710–770.

[14] Jones, A. and M. Sergot, *A formal characterisation of institutionalised power*, Journal of IGPL **4** (1996), pp. 427–443.

[15] Kulicki, P., R. Trypuz and M. Sergot, *Who is obliged when many are involved? Labelled transition system modelling how obligation arises*, Artificial Intelligence and Law **29** (2021), pp. 395–415.

[16] Lewinski, M., *Practical argumentation as reasoned advocacy*, Informal Logic **37** (2017), pp. 85–113.

[17] Lifschitz, V., *What is answer set programming?*, Proceedings of the AAAI Conference on Artificial Intelligence (2008).

[18] Lindahl, L., "Position and Change: A Study in Law and Logic," Synthese Library, Springer, 1977.

[19] Makinson, D., *On the formal representation of rights relations*, Journal of Philosophical Logic **15** (1986), pp. 403–425.

[20] Markovich, R., *No match-making but biconditionals. agents and the role of the state in legal relations*, in: A. Rotolo, editor, *Legal Knowledge and Information Systems 2015 – Proceedings of the 29th JURIX Conference. Frontiers in Artificial Intelligence and Applications*, IOS Press, 2015 pp. 161–164.

[21] Markovich, R., *Understanding Hohfeld and formalizing legal rights: the Hohfeldian conceptions and their conditional consequences*, Studia Logica **108** (2020), pp. 129–158.

[22] Markovich, R. and O. Roy, *A logical analysis of freedom of thought*, in: P. P. Fenrong Liu, Alessandra Marra and F. V. D. Putte, editors, *Deontic Logic and Normative Systems DEON 2020/21 proceedings*, College Publications, 2021 pp. 245–260.

[23] Ogilvie, D. M. and K. M. Rose, *Self-with-other representations and a taxonomy of motives: two approaches to studying persons*, Journal of personality **63** (1995), pp. 643–79.

[24] Pascucci, M. and G. Sileno, *A formal, diagrammatic and operational study of normative relations*, Journal of Logic and Computation (2023), pp. 1–30.

[25] Russell, S. J. and P. Norvig, "Artificial Intelligence: a modern approach," Pearson, 2009, 3 edition.

[26] Sartor, G., *Fundamental legal concepts: A formal and teleological characterisation*, Artificial Intelligence and Law **14** (2006), pp. 101–142.

[27] Sergot, M., *Normative positions*, Handbook of deontic logic and normative systems **1** (2013), pp. 353–406.

[28] Sharifi, S., A. Parvizimosaed, D. Amyot, L. Logrippo and J. Mylopoulos, *Symboleo: Towards a Specification Language for Legal Contracts*, in: *IEEE International Requirements Engineering Conference (RE'20), RE@Next! track*, 2020.

[29] Sileno, G., A. Boer and T. van Engers, *Towards a computational theory of action, causation and power for normative reasoning*, Proceedings of the 32nd International Conference on Legal Knowledge and Information Systems (JURIX 2019) (2019).

[30] Sileno, G. and M. Pascucci, *Disentangling deontic positions and abilities: a modal analysis*, in: F. Calimeri, S. Perri and E. Zumpano, editors, *Proceedings of the 35th Edition of the Italian Conference on Computational Logic (CILC 2020)*, 2020, pp. 36–50.

[31] van Binsbergen, L. T., L.-C. Liu, R. van Doesburg and T. van Engers, *eFLINT: a Domain-Specific Language for Executable Norm Specifications*, in: *GPCE 2020: Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming*, 2020.

[32] von Wright, G. H., *Is and ought*, in: M. C. Doeser and J. N. Kraay, editors, *Facts and Values: Philosophical Reflections from Western and Non-Western Perspectives*, Springer, Dordrecht, 1986 p. 31–48.

[33] von Wright, G. H., *Is there a logic of norms?*, Ratio Juris **4** (1991), p. 265–283.

[34] Walton, D., *Evaluating practical reasoning*, Synthese **157** (2007), pp. 197–240.