A formal, diagrammatic, and operational study of normative relations

Pascucci, M.; Sileno, G.

DOI
10.1093/logcom/exad011

Publication date
2023

Document Version
Final published version

Published in
Journal of Logic and Computation

License
CC BY

A formal, diagrammatic, and operational study of normative relations

MATTEO PASCUCCI, *Institute of Philosophy of the Slovak Academy of Sciences, v.v.i., Klemensova 19, 811 09 Bratislava, Slovakia.*
E-mail: matteo.pascucci@savba.sk

GIOVANNI SILENO, *Informatics Institute, University of Amsterdam, Science Park 900, 1098 XH Amsterdam, The Netherlands.*
E-mail: g.sileno@uva.nl

Abstract

In this work, we provide an extensive analysis of Hohfeld's theory of normative relations, focusing in particular on diagrammatic structures. Our contribution is threefold. First, we specify an extensional formal language to represent the main notions in the two families of normative relations identified by Hohfeld (i.e. the deontic and the potestative family). Our primary focus is on the part of the theory concerning potestative relations. In this regard, we assign a key role to the concept of ability, which is treated as a primitive notion and used to formulate three fine-grained definitions of power (outcome-centered, change-centered and force-centered). Second, on the basis of these definitions we build Aristotelian diagrams of opposition for deontic and potestative relations, improving, extending and systematizing previous proposals formulated in the literature. Third, we present a model-theoretic interpretation and a logic programming (ASP) implementation of the proposed framework, elaborating on the procedural dimension of normative reasoning.

Keywords: Hohfeldian relations, Aristotelian diagrams, diagrams of opposition, deontic relations, potestative relations, formalization, logic programming

1 Introduction

A core component of legal reasoning is understanding which *normative relations* (i.e. relations having normative relevance) hold among the various parties that are involved in a scenario, a challenge that requires in turn these relations to be appropriately defined [14, 15, 26, 40]. Suppose that we are analysing a scenario in which it is acknowledged that an amount of money has to be transferred in the light of a contractual agreement between two parties p and q; this scenario can be described either in terms of a duty of p towards q or in terms of a (claim) right of q against p. As the example shows, normative relations aim to set up expectations holding within a certain social setting, and so, at the very least, they exhibit a socially distributed nature when expressed in concrete forms.

The conceptual structure of normative relations has been widely investigated since modern legal scholarship (Leibniz, Bentham,...), but elements of the inquiry can be found also in ancient jurisprudence (e.g. Gaius's Institutes). A central position within this discussion is generally attributed to the seminal work by W. N. Hohfeld [14, 15]—although he plausibly drew significant inspiration from the work of J. Salmond [10, 31]. Hohfeld's declared objective was to study 'the basic conceptions of the law—the legal elements that enter into all types of jural interests' [14, p. 20] by analyzing the actual activity in courts, and so he supported a shift of focus from metaphysical (Leibniz) or law-maker concerns (Bentham, Austin) to 'concrete problems of litigation' [14, p. 58]. This perspective turns out to be particularly relevant in computational contexts (e.g. for digital contracts) due to the

Vol. 33, No. 4,
Advance Access Publication on 3 April 2023 https://doi.org/10.1093/logcom/exad011

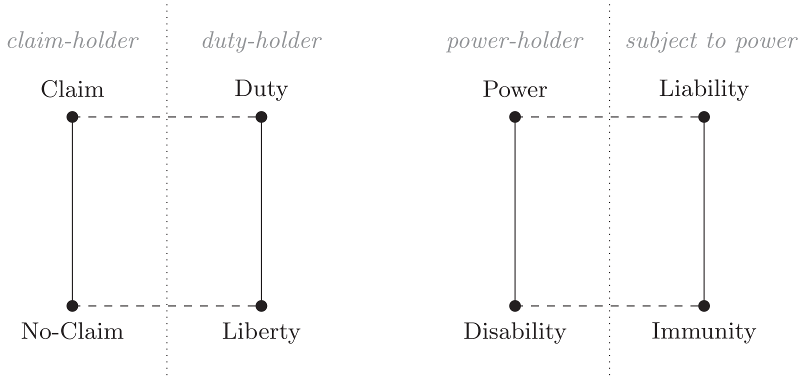

FIGURE 1. The two Hohfeldian squares: the one on the left is the deontic square, (first-order relations), the one on the right is the potestative square (second-order relations). A vertical axis divides each figure into two components representing the perspective of a certain category of normative parties.

instrumental nature of technology, more proximate to socio-institutional operationalization concerns than the substantive rights views discussed in legal doctrine.

Hohfeldian relations Hohfeld's theory of normative relations is graphically represented in terms of two diagrams, which have become known as the *Hohfeldian squares*. Such an illustration, here reproduced in Figure 1, unveils the dichotomy at the core of the theory. Hohfeld distinguishes between a first family of normative relations, which consists of 'duty', 'claim', 'liberty' (in the original terminology 'privilege') and 'no-claim', and a second family of normative relations, which consists of 'power', 'liability', 'disability' and 'immunity'. Relations of the first family are known as *first-order* or *deontic relations*; those of the second family are known as *second-order* or *potestative relations*. All these relations can change over time; actually, second-order relations are meant to capture how actions performed by some party may change normative relations holding upon the acting party or some other one(s). For instance, a party p may at some point utilize her power to create a duty upon a party q.

Relations analysed by Hohfeld generally involve *two normative parties and a certain behaviour*. A party can be either an individual, a group of individuals, an organization, an institution, etc.; a behaviour can be equated with a type of action (more generally, with a sequence of actions). All such relations can be also expressed in terms of modal verbs. For instance, in this context, saying that p has a duty towards q of performing A is the same as saying that p ought to perform A with regard to q, i.e. that q has normative expectations that p will perform A. These relations can be then conceived of as *ternary modalities* and one can accordingly speak of *Hohfeldian modalities* [28].

Contributions While Hohfeld's squares play an important role in legal education (see, e.g. Nyquist [25]), and as such they constitute a fundamental reference point to the theory of normative relations, their formal interpretation is not straightforward; consequently, they cannot be directly used for the purpose of a rigorous characterization of the theory.

In the present article we provide a threefold contribution: (i) a *formal analysis* of Hohfeldian relations in terms of an extensional language, (ii) a graphical characterization of their deductive behaviour in terms of *diagrams of opposition* (also known as Aristotelian diagrams) and (iii) an

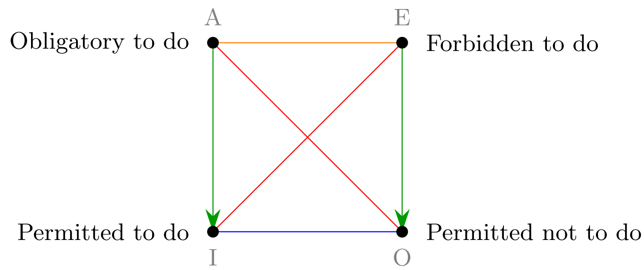

FIGURE 2. Aristotelian square of opposition for basic deontic modalities, with subaltern (green), contrary (orange), sub-contrary (blue) and contradictory (red) bindings, and the traditional labels associated to each node (A, E, I, O). Colours are displayed in the online version of this work.

interpretation of the framework via model-theoretic semantics and an implementation via Answer Set Programming (ASP) for automated normative reasoning.[1]

As far as part (i) is concerned, we opt for a language in which the relations at issue can be treated as ternary predicates. This is a variation of the language of classical logic where individual constants and individual variables are typed. Furthermore, in our language, we assign a central role to a predicate representing the notion of 'ability': such a predicate is the only one taking formulas (and other terms) as arguments and is used in three definitions of the notion of 'power', which will be called outcome-, force- and change-centered power, respectively. We will see that this choice allows for a significantly more fine-grained account of potestative relations than the one originally offered by Hohfeld. Moreover, we will highlight an asymmetry in the mainstream formal rendering of first-order and second-order Hohfeldian relations, which follows the approach by Lindahl [21] and is due to the use of negation in certain definitional clauses.

As far as part (ii) is concerned, we use diagrams of opposition (Aristotelian diagrams) to capture how Hohfeldian relations logically interact within each of the two families. An Aristotelian diagram is a geometrical representation of a theory based on four relations called contrariety, contradiction, sub-contrariety and subalternation; it generalizes the idea behind (Aristotelian) *polygons of opposition*, which have been widely used in linguistic, literary and semiotic studies, and have recently attracted a renewed interest in formal logic [5, 9, 30]. Aristotelian diagrams are different from the diagrams originally proposed by Hohfeld; yet, their choice in a logical setting is convenient since they fare much better in terms of clarifying the logical connections among the notions involved: basically, an Aristotelian diagram can be entirely described in terms of a set of statements and connections between the truth-conditions of pairs of them. As an illustration, the most known Aristotelian square of opposition for *basic* deontic modalities, i.e. obligation, permission and prohibition (also referred to as forbiddance), is here reproduced in Figure 2.

As far as part (iii) is concerned, we present model-theoretic semantics, which differs from the standard semantics for classical first-order logic due to the use of multiple domains of objects, among which a domain of formulas-as-objects. Moreover, we provide a logic programming implementation

[1] This work is an extended version of Pascucci and Sileno [28], where the ideas behind part (i) and part (ii) of the contribution were laid down in a concise way. The formal language employed undergoes a deeper technical analysis than the one sketched in [28] and is associated with a model-theoretic semantics that was not included in the original presentation (section 6). Aristotelian diagrams are here introduced in a mathematical setting, and their properties are discussed in a detailed way (section 2). The ASP implementation of the framework (section 7) is entirely new.

(namely in ASP [20]) of the proposed conceptualization, together with an application for normative exploration, integrating the Event Calculus [19, 34] with additional features to deal with procedural aspects of normative reasoning. The resulting code is publicly available[2] and can be readily embedded in computational agents.

Related works The literature on normative concepts and normative systems is vast and spans across various disciplines (e.g. [16, 17, 21–23, 33]). In general, the deontic dimension of normative reasoning has traditionally attracted most of the attention in the literature (e.g. Alchourrón [1]). For the aims of the present work, we refer here only to the much fewer works attempting to bridge normative concepts and diagrams of opposition, citing more general works technically relevant in the main text when needed. The possibility of studying logical relations among deontic statements via diagrams of opposition was already suggested in ancient times, it was revisited by Leibniz, and more recently by Blanché [6]. An earlier and informal geometrical analysis of Hohfeld's theory in terms of Aristotelian squares of opposition is due to O'Reilly [26], who extended Sumner's analysis [40] introducing concepts that will be here formally revisited to build a *change-centered* potestative square. Sileno et al. [36] argued for the construction of Hohfeldian prisms, based on triangles of opposition, and introduced a *force-oriented* potestative square. More recently, in an independent effort parallel to our work, de Oliveira Lima et al. [7] presented a review of the various accounts of theories of opposition applied to normative relations. Their work, giving generally more space to deontic concepts, elaborates further on the connections between weak and strong forms of permission, but provides also arguments on the constitutive nature of power. The potestative diagram they propose at the end of their paper (focusing on the competence necessary to modify a legal position) can be related to the *change-centered* one of our framework.

With respect to the implementation part of the paper, the reader can find an introduction to ASP in [20]. ASP is a declarative programming paradigm based on a stable-model semantic [12] and oriented towards difficult (NP-hard) search problems, which is increasingly used to model and solve problems in research and industry in a wide range of application domains. Not surprisingly, it has also been used in contributions in the normative systems literature, as for instance in the *InstAL* framework ('An Institutional Action Language') [27], used e.g. in [8] to provide semantics to ODRL (Open Digital Rights Language), a W3 recommendation policy expression language.[3] In the present work, we will build upon the connection between institutional power and Event Calculus [19, 34] formulated in [37].

Outline of the article The paper proceeds as follows. Section 2 provides some formal background on Aristotelian diagrams. Section 3 presents the notation upon which our proposal will be constructed, and introduces our formalization of the normative concepts illustrated in Hohfeld's framework. Section 4 investigates a reinterpretation of Hohfeldian relations in terms of Aristotelian diagrams, reorganizing, integrating and extending the various contributions presented in the literature. Section 5 argues that, in order to obtain a proper symmetric treatment of the two families of Hohfeldian modalities (in particular, with respect to the notions of liberty and of power), one has to move from squares to hexagons of opposition. Section 6 provides a model-theoretic semantics to interpret the proposed formal language, thus paving the way to the development of specific logical systems for normative reasoning based on our framework. Finally, section 7 presents a logic programming implementation of the proposed conceptualization, and an example of its utilization for exploring a given normative model. A note on future developments ends the paper.

[2] https://github.com/gsileno/normative-primitives-asp.
[3] https://www.w3.org/TR/odrl-model/

2 Aristotelian diagrams

2.1 Definitions

Aristotelian diagrams are geometrical figures composed of *vertices*, which are associated with statements (sometimes expressed by formulas of a formal language), and (possibly directed) *edges*, which are associated with *Aristotelian relations*[4]:

DEFINITION 2.1 (Aristotelian Relation).
An *Aristotelian relation* between two statements ϕ and ψ is a relation called either *subalternation* or *contrariety* or *sub-contrariety* or *contradiction*, where:

- ψ is a subalternate of ϕ iff the truth of ϕ entails the truth of ψ, whereas the truth of ψ does not entail the truth of ϕ;
- ϕ and ψ are contraries iff at most one between ϕ and ψ can be true;
- ϕ and ψ are sub-contraries iff at most one between ϕ and ψ can be false;
- ϕ and ψ are contradictories iff ϕ is true precisely when ψ is false.

DEFINITION 2.2 (Aristotelian Diagram).
Given a finite set of formulas Γ, an Aristotelian diagram over Γ is a graph whose vertices are labelled by elements of Γ. Each vertex v is labelled by a distinct formula. An edge e of the graph connecting two vertices v_1 and v_2 is associated with an *Aristotelian relation* R_e between the formulas labelling v_1 and v_2.

An implicit assumption behind Aristotelian diagrams is that edges always connect distinct vertices, which reflects the fact that any Aristotelian relation is, by Def. 2.1, irreflexive. Also, according to Def. 2.1, contrariety, sub-contrariety and contradiction are symmetric relations. By contrast, subalternation is not a symmetric relation: in cases in which the truth of ϕ entails the truth of ψ and *vice versa* we rather say that ϕ and ψ are *logically equivalent*. In the graphical representation of a diagram, it is common to adopt the convention that symmetric relations are associated with non-directed edges.

2.2 Tracking Aristotelian diagrams

The most common sort of Aristotelian diagram is the *Aristotelian square*, which can be described as a square whose vertices v_1, v_2, v_3 and v_4 are respectively labelled by statements ϕ_1, ϕ_2, ϕ_3 and ϕ_4, and such that Aristotelian relations are associated with pairs of its vertices as follows:

- (v_1, v_2) represents contrariety;
- (v_1, v_4) and (v_2, v_3) represent contradiction;
- (v_3, v_4) represents sub-contrariety;
- (v_1, v_3) and (v_2, v_4) represent subalternation (the second element of each pair is a subalternate of the first).

[4]The following definitions of an Aristotelian relation and of an Aristotelian diagram are taken from [29]; for further technical discussion on Aristotelian relations and polygons of opposition, see [9].

The aforementioned relations reflect how the statements ϕ_1, ϕ_2, ϕ_3 and ϕ_4 are logically related *in a specified setting*. Thus, when building an Aristotelian square one typically starts by choosing an underlying logical setting, which can be either a formal theory (e.g. a logical system S) or an informal theory used in ordinary reasoning, and then identifies a set Γ of four formulas that are suitably related in such a setting. For instance, a known method for doing this employs quantifiers (e.g. 'everyone is wise' vs. 'someone is wise') and the distinction between *internal negation* (e.g. 'everyone is unwise', which may also be rewritten as 'everyone is not wise') and *external negation* (e.g. 'not everyone is wise'). Alternatively, one can employ oppositions between modal notions, together with the two kinds of negation. This strategy is particularly relevant for normative reasoning, as Figure 2 illustrates. For instance, in a formal theory of deontic logic with action types, given the formula $Obl(A)$ ('it is obligatory to perform an action of type A'), one can specify external negation via an operator of sentential negation and get $\neg Obl(A)$ ('it is not obligatory to perform an action of type A'). Furthermore, one can specify an internal negation concerning action types with a complementation operator and get the formula $Obl(\overline{A})$, which reads as 'it is obligatory to perform an action of a type different from A' (possibly including an *idle* action), and can be arguably regarded as synonymous with 'it is forbidden to perform an action of type A', denoted as $Forb(A)$. Finally, one can take both sorts of negation to form the dual of the initial formula, i.e. $\neg Obl(\overline{A})$, which means 'it is not obligatory to perform an action of a type different from A'. The latter can be regarded as synonymous with 'it is permitted to perform an action of type A', denoted as $Perm(A)$. With these elements, one can track the four Aristotelian relations among basic deontic modalities relying on a sufficiently strong logical system and obtain the following exhaustive list of relations:

- $Obl(A)$ and $Obl(\overline{A})$ are contraries;
- $\neg Obl(\overline{A})$ is a subalternate of $Obl(A)$, and $\neg Obl(A)$ of $Obl(\overline{A})$;
- $\neg Obl(\overline{A})$ and $\neg Obl(A)$ are sub-contraries;
- $Obl(A)$ and $\neg Obl(A)$ are contradictories, as well as $Obl(\overline{A})$ and $\neg Obl(\overline{A})$.

Generalizing this approach, a possible way of developing a square of opposition starting with a statement ϕ is the following. One needs:

1. to express external negation in terms of an operator for sentential negation (not necessarily Boolean negation) $\sim$ applied to ϕ (so as to get $\sim\phi$) and giving rise to contradiction;
2. to identify a device able to express internal negation and get a formula ψ;
3. to test the formulas $\sim\phi$ and $\sim\psi$ for sub-contrariety.

Once this is done, the additional Aristotelian relations between pairs of formulas in the set $\Gamma = \{\phi, \psi, \sim\phi, \sim\psi\}$ that are needed to build the square are obtained automatically.

Aristotelian squares will play a central role in our exposition (see, e.g. Figure 3). We will also work with *Aristotelian hexagons*, which are obtained by adding to a square two vertices v_5 and v_6, respectively labelled by formulas ϕ_5 and ϕ_6 and s.t.:

- (v_1, v_5), (v_2, v_5), (v_6, v_3) and (v_6, v_4) represent subalternation (the second element of each pair is a subalternate of the first).

A method for building Aristotelian hexagons consists in taking ϕ_5 to be the disjunction of ϕ_1 and ϕ_2, as well as in taking ϕ_6 to be the conjunction of ϕ_3 and ϕ_4. Finally, we will combine squares and hexagons so as to get more complex Aristotelian diagrams, by tracking relations between pairs of vertices originally belonging to different figures (see, e.g. Figure 4).

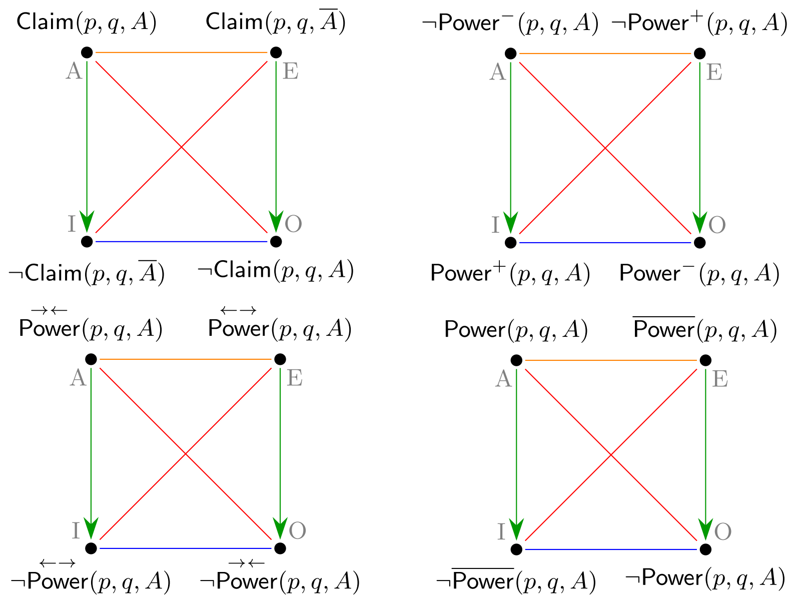

FIGURE 3. Deontic and potestative (change-centered, force-centered and outcome-centered) Aristotelian squares of opposition constructed from the two Hohfeldian diagrams. The usual convention on the colour of bindings apply.

3 Formalization

Several proposals to represent the theory of Hohfeldian relations in a formal setting have been outlined over the years. Among the early approaches one finds Anderson's idea of employing formulas with three parameters: two of which standing for normative parties and the third one standing for a certain behaviour [2]. This idea has influenced many of the subsequent approaches. Two systematic proposals based on the logic of agency are due to Lindahl [21] and Makinson [22], who also provide a taxonomy of normative relations that can be expressed within their formal frameworks, by combining the various parameters used. Recently, the interest in formalizing Hohfeld's theory has been renewed by investigations in the area of modal logic (in this regard, see, for instance, the discussion offered by Markovich [23], Sileno and Pascucci [38] or Novotná and Pascucci [24]).

A central insight behind formalization, which appears to be systematically exploited e.g. by Lindahl [21], is that all Hohfeldian relations are *family-wise interdefinable*, in the following sense: one can take a single deontic relation and a single potestative relation as primitives and, for each family, introduce all the other relations thanks to a small set of logical operations. Here we adhere to this view, although making some further steps with respect to conceptual clarification in the case of second-order relations, since we take all of them to be defined, in various ways, in terms of the more fundamental notion of 'ability', further developing ideas that we presented in [28, 36–38].

In its wider sense, we take ability as the possibility of an agent to causally intervene on her environment (i.e. producing some transition) by some behavioural means [38]. In a normative framing, power can then be seen as *institutional ability* ascribed to agents (i.e. an ability intervening on institutional objects, such as normative relations), because its enactment produces changes in the institutional realm. More precisely, relations of the potestative family concern *actions* that trigger changes of first-order or even second-order normative relations[5]; for instance, an action b creating a duty on a party q to perform an action a for a party p. A possible way of writing that p has such a power is by means of the formula (cf. the predicate *has_ability* investigated in [38]):

$$\mathsf{Ability}(p, b, \psi),$$

where ψ is, for instance, a Hohfeldian relation among p, q and an action of type a that is issued at p's performance of an action of type b. To illustrate this, the case in which the triggered Hohfeldian relation is a 'claim' can be expressed as:

$$\mathsf{Ability}(p, b, \mathsf{Claim}(p, q, a)).$$

The language we will utilize gives a central role to expressions of this type.

3.1 Language

Our formal setting to represent Hohfeldian relations will be based on an extensional language with typed variables and constants. This language will be denoted by $\mathcal{L}$ throughout the presentation.

DEFINITION 3.1 (Vocabulary).
The vocabulary of $\mathcal{L}$ consists of the following symbols (all mentioned sets are taken to be non-empty and pairwise disjoint):

- a set $\mathtt{Var}_{np}$ of individual variables (denoted by $x, y, z, ...$), which represent *normative parties*;
- a set $\mathtt{Var}_{at}$ of individual variables (denoted by $\alpha, \beta, \gamma, ...$), which represent *action types*;
- a set $\mathtt{Const}_{np}$ of individual constants (denoted by $p, q, r, ...$), which represent *normative parties*;
- a set $\mathtt{Const}_{at}$ of individual constants (denoted by $a, b, c, ...$), which represent *action types*;
- a set $\mathtt{Pred}$ of ternary predicate constants;
- the binary predicate constant $\asymp$ for identity (of normative parties, and of action types);
- the symbol $-$ (overline), which represents the *complement* of an action type;
- the logical operators $\neg$ (Boolean negation) and $\rightarrow$ (material implication);
- the quantifier $\forall$;
- round brackets.

One of the elements of $\mathtt{Pred}$ is the predicate constant $\mathsf{Ability}$, which will be analysed in a separate way from the others, since it plays a special role in our formalism. Hereafter, we take $\mathtt{Var} = \mathtt{Var}_{np} \cup \mathtt{Var}_{at}$ to denote the set of *individual variables* and $\mathtt{Const} = \mathtt{Const}_{np} \cup \mathtt{Const}_{at}$ to denote the set of *individual constants*. Moreover, let $\mathtt{Term}_{np} = \mathtt{Var}_{np} \cup \mathtt{Const}_{np}$ denote the set of *terms for normative parties* and $\mathtt{Term}_{at} = \mathtt{Var}_{at} \cup \mathtt{Const}_{at}$ the set of *terms for action types*. Finally, let $\mathtt{Const}^*_{at}$ be the closure of $\mathtt{Const}_{at}$ under action complementation, which can be iterated an arbitrary number of times, and $\mathtt{Term}^*_{at} = \mathtt{Const}^*_{at} \cup \mathtt{Var}_{at}$.

Moreover, throughout the rest of the exposition, we will adopt the following *convention on the symbols used* (for the sake of convenience, we choose a notation for our formal language that adheres to the subsequent notation used in the ASP implementation, although it differs from standard notations for classical logic):

[5] Although for most legal scholars legal power concerns only first-order relations, see footnote 8.

- P denotes an arbitrary predicate in $\texttt{Pred} \setminus \{\mathsf{Ability}\}$;
- X, Y, Z denote arbitrary terms in $\texttt{Term}_{np}$;
- A, B, C denote arbitrary terms in $\texttt{Term}^*_{at}$;
- V denotes an arbitrary variable in $\texttt{Var}$.

This convention will allow us to avoid making reference to quantification over relevant sets of symbols in the definitions below and in the subsequent parts of the article. Sometimes, we will use A, B, C, etc. with the additional specification that they denote elements of $\texttt{Const}^*_{at}$ only, rather than elements of the larger set $\texttt{Term}^*_{at}$, as they generally do.

DEFINITION 3.2 (Formulas).
Formulas of $\mathcal{L}$ are specified by the grammar below, where ψ has either the form $P(X, Y, A)$ or the form $\neg P(X, Y, A)$:

$$\phi ::= P(X, Y, A) \mid X \asymp Y \mid A \asymp B \mid \mathsf{Ability}(X, A, \psi) \mid \neg\phi \mid \phi \to \phi \mid \forall V\phi.$$

Additional logical operators, such as conjunction ($\wedge$), disjunction ($\vee$), material equivalence ($\leftrightarrow$) and the particular quantifier ($\exists$), can be defined in the usual way. $X \not\asymp Y$ and $A \not\asymp B$ are shorthands for $\neg(X \asymp Y)$ and $\neg(A \asymp B)$, respectively. The notions of *free* and *bound occurrence of a variable*, as well as the notion of a *closed formula* (a formula in which all occurrences of variables are bound) are defined as in classical first-order logic [13].

Formulas involving the predicate $\mathsf{Ability}$ indicate that language $\mathcal{L}$ has expressive resources that go beyond those available in a (typed) language of classical first-order logic, since one of the arguments of $\mathsf{Ability}$ is a formula. Yet, given that there is no quantification on formulas or properties, formulas appearing as arguments of $\mathsf{Ability}$ can be ultimately regarded as a third sort of individual constants, in addition to $\texttt{Const}_{np}$ and $\texttt{Const}_{at}$. Moreover, $\mathcal{L}$ is an extensional language, since modal notions are all treated as predicates.[6]

Note that complementation is the only operation that allows one to build *complex* action types: given a term for an action type A, $\overline{A}$ denotes the complement of A, i.e. the type of any action that does not instantiate A. For instance, if A is the action type 'paying a rental fee', then any performed action that does not instantiate this type is taken to instantiate the type $\overline{A}$. We will present different ways of building logical systems over $\mathcal{L}$; in all these systems, we will assume the deductive principles of the Predicate Calculus with identity, together with the two axioms $A \asymp \overline{\overline{A}}$ and $A \not\asymp \overline{A}$, where $A \in \texttt{Const}^*_{at}$. From all these one further principle regulating the use of action complementation follows, namely the Law of Double Complementation, which can be expressed via the schema

$$\phi \leftrightarrow \phi[\overline{\overline{A}}//A]$$

and whose meaning is that any formula ϕ is provably equivalent to the formula obtained from it by replacing some occurrences of term A with term $\overline{\overline{A}}$.[7]

The next step is illustrating *definitional procedures* to obtain all notions within each of the two Hohfeldian families. A preliminary methodological remark concerning our exposition can serve as

[6] The language $\mathcal{L}$ is meant to be a rigorous support for the investigation of the diagrammatic theories presented in the paper. To cover more general normative mechanisms, however, the predicate $\mathsf{Ability}$ should be allowed in ψ in Def. 3.2; this is left to future investigation and preliminarily explored in the ASP implementation.

[7] In axiomatic bases of the Predicate Calculus with identity (see, e.g. [4]), any two constants c and c' that are provably identical can be replaced in any formula, since from the theorems $c \asymp c'$ and $(c \asymp c') \to (\phi \leftrightarrow \phi(c'//c))$ one obtains $\phi \leftrightarrow \phi(c'//c)$. In the approach proposed in [28], the identity predicate is not employed and the Law of Double Complementation is taken as a principle to be added directly to axiomatic bases.

a guide through sections 3.2 and 3.3: definitional equivalences will be stated in terms of formulas involving terms for normative parties (X, Y, etc.) and terms for action types (A, $\overline{A}$, etc.). By contrast, specific formulas used as hypotheses or examples will involve individual constants for normative parties (p, q, etc.) and terms for action types (since we will discuss examples involving specific individuals and possibly complex types of action). Finally, from time to time, we will consider the possibility of developing logical systems over the formal language $\mathcal{L}$ and we will mention axiom-schemata for these systems, which involve universal quantification over sets of variables (yet, these schemata can be instantiated thanks to the deductive principles of the Predicate Calculus, which we always take for granted).

A model-theoretic semantics for language $\mathcal{L}$ and theories built over it will be presented in section 6. Before doing that, we will engage in a thorough syntactic and diagrammatic analysis. We will illustrate: how language $\mathcal{L}$ can be used to formalize Hohfeldian relations (remainder of section 3); how, on the basis of the formalization adopted, one can transform Hohfeldian squares into Aristotelian squares of opposition labelled with $\mathcal{L}$-formulas (section 4); how one can preserve the symmetry of Hohfeldian squares in the new sort of diagrams (section 5).

3.2 First-order or deontic Hohfeldian relations

According to the idea that Hohfeldian relations belonging to the same family are all interdefinable, we will take the notion of 'claim' as primitive, represented via the predicate symbol Claim and show how to introduce the other first-order notions. We will do this via a definitional procedure that follows the approach by Lindahl [21]. Thus, let us assume that $\mathsf{Claim} \in \mathtt{Pred}$. Given two normative parties p and q and an action type A, we start with the sentence that p has a claim on q that A be performed, expressed by $\mathsf{Claim}(p,q,A)$. The other three formulas that we want to get at the end of the definitional procedure are the following: $\mathsf{Liberty}(p,q,A)$, $\mathsf{Duty}(p,q,A)$ and $\mathsf{NoClaim}(p,q,A)$. One can immediately notice that the last Hohfeldian relation in this list is just the negation of the initial assumption. As a matter of fact, $\mathsf{NoClaim}(p,q,A)$ can be obtained from the initial statement thanks to the definitional schema below, which exploits Boolean negation:

$$\mathsf{NoClaim}(X,Y,A) =_{def} \neg\mathsf{Claim}(X,Y,A).$$

Furthermore, one can treat 'claim' and 'duty' as *correlative* notions, in the sense that they are two faces of the same modality, seen from the points of view of the two normative parties involved. Therefore, $\mathsf{Duty}(p,q,A)$ just results from $\mathsf{Claim}(p,q,A)$ via a permutation of the two normative parties, according to the schema below:

$$\mathsf{Duty}(X,Y,A) =_{def} \mathsf{Claim}(Y,X,A).$$

Finally, one can define 'liberty' in terms of 'claim', Boolean negation, a permutation of normative parties and action complementation, getting $\mathsf{Liberty}(p,q,A)$ via the schema below:

$$\mathsf{Liberty}(X,Y,A) =_{def} \neg\mathsf{Claim}(Y,X,\overline{A}).$$

Note that in any logical system S closed under the Law of Double Complementation, the definitional equivalences presented so far yield the following theorem:

$$\neg\mathsf{Duty}(X,Y,A) \leftrightarrow \mathsf{Liberty}(X,Y,\overline{A})$$

i.e. the negation of a duty of performance corresponds to the liberty of non-performance.[8] In a more refined representation of specific Hohfeldian relations, one can also take into account the statement that the two normative parties involved are typically distinct. For instance: $\mathsf{Duty}(p, q, A) \wedge p \not\asymp q$.

3.3 *Second-order or potestative Hohfeldian relations*

For the sake of uniformity with our treatment of first-order Hohfeldian relations, and to align with the original Hohfeldian framework, we want to obtain formal representations of 'power' that involve two main normative parties: a power-holder p and a power-subject q, i.e. a party that is required to act in light of the creation of a 'claim'. This will be said to be a *canonic power*[9] construct with reference to a given action of type A and will be denoted as $\mathsf{Power}(p, q, A)$. The reading of the latter expression is given by the definitional schema below:

$$\mathsf{Power}(X, Y, A) =_{def} \exists\beta(\mathsf{Ability}(X, \beta, \mathsf{Claim}(X, Y, A))),$$

where X is the power-holder, and Y is the party subject to power.[10] Thanks to this move, the four fundamental potestative relations in Hohfeld's theory can then be encoded following the same syntactic pattern used in the case of first-order relations, namely treating all of them as ternary relations whose first and second arguments are normative parties and whose third argument is an action type (i.e. as the definitional schema above indicates, the action type mentioned in the relation affected).

Starting from the statement that party p has (canonic) power over party q with respect to action type A, expressed by $\mathsf{Power}(p, q, A)$, we will now obtain three related formulas: $\mathsf{Liability}(p, q, A)$, $\mathsf{Disability}(p, q, A)$ and $\mathsf{Immunity}(p, q, A)$. First, one can treat 'disability' as the negation of 'power' and get $\mathsf{Disability}(p, q, A)$ via the following definitional schema:

$$\mathsf{Disability}(X, Y, A) =_{def} \neg\mathsf{Power}(X, Y, A).$$

Then, one can observe that 'liability' and 'power' are correlatives, i.e. the same normative relation seen from two different perspectives (like 'duty' and 'claim'). Thus, one can get $\mathsf{Liability}(p, q, A)$ via a definitional schema involving a permutation of the two normative parties:

$$\mathsf{Liability}(X, Y, A) =_{def} \mathsf{Power}(Y, X, A).$$

Finally, one can treat 'immunity' as the absence of 'power' with respect to a permutation of the two normative parties, thus getting $\mathsf{Immunity}(p, q, A)$ from the schema:

$$\mathsf{Immunity}(X, Y, A) =_{def} \neg\mathsf{Power}(Y, X, A).$$

In any logical system S closed under the Law of Double Complementation, the last two equations yield the following theorem:

$$\neg\mathsf{Liability}(X, Y, A) \leftrightarrow \mathsf{Immunity}(X, Y, A)$$

[8] Note that the definition of liberty of q towards p of performing A as $\mathsf{Liberty}(q, p, A)$ is weaker than the normative position we usually associate to people deemed free to behave as they prefer, as the first definition holds also when someone has the duty to A. Section 5 will further clarify this point (half-liberty vs. full-liberty).

[9] In the words of Kocourek: '[..] power (*Können*) is a legal concept only in-so-far as it includes within its ambit, claims or duties' [18].

[10] Note that in general, there may be three parties for which the creation of a duty is relevant: a power-holder X, a (potential) duty-holder Y and a (potential) claim-holder Z. The canonic form of power reifies the plausible assumption that power-holders enact their power to become claimant (thus, X and Z are the same).

i.e. the negation of a liability (correlatively, power) towards performance corresponds to the immunity (disability) towards performance. In contrast, in the case of first-order relations, the negation of a duty of performance corresponded to a liberty of *non*-performance. This means that, according to this formalization, which is inspired by Lindahl's proposal in [21], as well as by analogous proposals presented by subsequent authors, e.g. [22, 23, 32], the two families of Hohfeldian relations seemingly lack the symmetry suggested by Hohfeld's diagrams (and that plausibly made the framework so successful). This somewhat unexpected result in the formal literature was one of the motivations behind our research.

4 Hohfeldian squares and Aristotelian squares

O'Reilly [26] (extending Sumner's analysis [40]) observes that, for any choice of a primitive deontic and potestative modality, one can build two Aristotelian squares of opposition whose vertices are labelled by a formula where only the primitive modality of the relevant family is mentioned. Indeed, as the previous formalization indicates, any of the four relations on each of the Hohfeldian squares can be defined in terms of any other relation belonging to the same family. Moreover, in our framework, the four formulas needed as labels for a square are obtained by the possible combinations of Boolean negation and action complementation (either both present, or both absent, or one present and one absent), which is a first step towards tracking Aristotelian diagrams (see section 2.2).

4.1 Deontic square of opposition

Let us choose the statement $\mathsf{Claim}(p,q,A)$ as an initial hypothesis in order to build a square of opposition extracted from the first-order Hohfeldian diagram. This choice, together with the idea of making use of all possible combinations of internal and external negation, leads to the following set of labels DR (for 'deontic relations'):

$$\mathsf{DR} = \{\mathsf{Claim}(p,q,A), \mathsf{Claim}(p,q,\overline{A}), \neg\mathsf{Claim}(p,q,\overline{A}), \neg\mathsf{Claim}(p,q,A)\}.$$

We will say that this is a Claim-based description of DR; exploiting the definitional equivalences in section 3.2 one could write, equivalently, a Duty-, $\mathsf{Liberty}$- or $\mathsf{NoClaim}$-based description.

Due to the properties of Boolean negation and the Law of Double Complementation, the set DR is almost sufficient to get a deontic square of opposition. The only part missing is a way of characterizing the *subalternation relation*, namely the fact that the truth of $\mathsf{Claim}(p,q,A)$ entails the truth of $\neg\mathsf{Claim}(p,q,\overline{A})$ (but not the other way around) and that the truth of $\mathsf{Claim}(p,q,\overline{A})$ entails the truth of $\neg\mathsf{Claim}(p,q,A)$ (but not the other way around).[11] As a general solution, this can be achieved by considering logical systems that contain the following axiom-schema:

$$\forall x \forall y\ (\mathsf{Claim}(x,y,A) \rightarrow \neg\mathsf{Claim}(x,y,\overline{A})).$$

This formula can be seen as corresponding to the $Obl(\phi) \rightarrow Perm(\phi)$ axiom-schema traditionally used in deontic logics and conveys the picture of ideal scenarios where norms associated to Hohfeldian relations do not conflict among them. A simple example: if Ada has a claim on Bill that he pays a rental fee, then she cannot have a claim that he does not do so (i.e. if Bill is obliged to pay a rental fee, he cannot be also prohibited from doing so; hence, he must be permitted to do so).

[11] The two subalternation relations and the contrary relation entail the sub-contrariety relation.

4.2 Change-centered potestative square of opposition

O'Reilly considers power as the ability of a party p to *affect* a party q with respect to a Hohfeldian relation. This can be rephrased in our formal setting by saying that there are triggering actions that produce a *change* w.r.t. a Hohfeldian relation expressed by a formula ψ. Focusing on the relation of 'claim', namely on cases where $\psi = \mathsf{Claim}(p, q, A)$, we can take a triggering action B and write the definitional equivalence for this O'Reillian notion of power as:

$$\begin{aligned}\mathsf{Power}_{OReilly}(X, Y, B, A) =_{def} \; & \mathsf{Ability}(X, B, \mathsf{Claim}(X, Y, A)) \\ & \vee \; \mathsf{Ability}(X, B, \mathsf{Claim}(X, Y, \overline{A})) \\ & \vee \; \mathsf{Ability}(X, B, \neg\mathsf{Claim}(X, Y, A)) \\ & \vee \; \mathsf{Ability}(X, B, \neg\mathsf{Claim}(X, Y, \overline{A})).\end{aligned}$$

Since reference to the specific triggering action is not fundamental to explicate this conceptual construction (i.e. the sort of power mentioned by O'Reilly arises as soon as *some* relevant triggering action is available), we can use quantification over the set of possible action types and the O'Reillian notion of power to define a form of *positive-change power* (Power^{+}) with respect to a claim involving a given action type A, as below:

$$\mathsf{Power}^{+}(X, Y, A) =_{def} \exists\beta(\mathsf{Power}_{OReilly}(X, Y, \beta, A))$$

At the end of this definitional procedure, we notice that positive-change power corresponds to the ability of affecting (in any sense) a given normative relation. O'Reilly also refers to a distinct form of internal negation, capturing the ability of a party p to *not* affect a party q with respect to a Hohfeldian relation. This will be said to represent a form of *negative-change* or *no-change power* (Power^{-}). In other words, this means that p may choose to behave in a way that does not produce any change with respect to a claim involving action type A. In our formalization, the resulting definitional equivalence would be:

$$\mathsf{Power}^{-}(X, Y, A) =_{def} \exists\beta\,(\neg\mathsf{Power}_{OReilly}(X, Y, \beta, A)).$$

We can illustrate the difference between positive-change and negative-change power with a simple example: suppose Emma lost a bag and that this bag was found by somebody and brought to the lost property office. By visiting that office and declaring to be the owner of the lost bag, she creates a claim on the office employees to give the bag back to her (i.e. there is an action available to her which gives rise to a O'Reillian form of power in the specified setting). In contrast, if she does not visit that office, she does not change any Hohfeldian relation involving office employees (i.e. there is an action available to her, which does not give rise to a O'Reillian form of power in the specified setting).

Starting from these concepts, we can define a set $\mathsf{PR}^{\pm}$ of potestative *change-centered* relations with the aim of building a square of opposition for second-order Hohfeldian relations. Here the four formulas needed are obtained via possible combinations of Boolean negation and of positive- vs. negative-change power:

$$\mathsf{PR}^{\pm} = \{\mathsf{Power}^{+}(p, q, A), \mathsf{Power}^{-}(p, q, A), \neg\mathsf{Power}^{-}(p, q, A), \neg\mathsf{Power}^{+}(p, q, A)\}.$$

To build a square of opposition upon $\mathsf{PR}^{\pm}$, and follow O'Reilly's approach, one has to add the principle below, which captures subalternation:

$$\forall x \forall y \, (\neg \mathsf{Power}^{-}(x,y,A) \to \mathsf{Power}^{+}(x,y,A)).$$

In contrast to the deontic case, such a principle is not independent of the rest: it is entailed by the proposed formalization, in the plausible assumption that the set of action types is non-empty.

4.3 Force-centered potestative square of opposition

The rewriting of the notion of power considered by O'Reilly in terms of 'ability' unveils that it is a rather complex concept. One may then wonder whether a square of opposition may be constructed starting from simpler forms of power. Different characterizations of actions exist in human language, mapping to different levels of abstraction [39], e.g. the *behavioural* or *procedural* characterization, relating to the action task or the *productive* characterization, relating to its outcome. We can similarly attempt to formulate different definitions of power constructed at different abstraction levels. Indeed, previous works argued that, at behavioural level, power relations can be put in analogy to force fields determining attraction, repulsion and absence of those (independence) at the occurrence of interventions ([36], [35, Ch.4]). To express such physical metaphor, we need to separate the *stimulus* component (a particular type of action, such as a verbal command) and the consequent target *manifestation* (e.g. a type of action that is due or expected on the basis of the stimulus, as in the concept of *pliance*). If the action-manifestation is denoted by the action type symbol A, then, the action-stimulus can be conveniently represented via the symbol "A" to emphasize the shared connection between signal and expected performance.

Technically, this means that we need to extend our language $\mathcal{L}$ with a *'stimulus' function* which, for every element A in the set $\mathtt{Term}^{*}_{at}$ received as input, produces the element "A" as output. Therefore, this yields a larger set of terms for action types, namely:

$$\mathtt{Term}^{*+}_{at} = \mathtt{Term}^{*}_{at} \cup \{\texttt{"A"} : A \in \mathtt{Term}^{*}_{at}\}.$$

Relevant scenarios can be then identified on, e.g. whether stimulus and manifestation converge (A is always performed in correspondence to its stimulus) or diverge (A is never performed in correspondence to its stimulus). Using our notation, the definitions of attraction and repulsion force-centered notion of power are:

$$\begin{aligned} \overset{\to\leftarrow}{\mathsf{Power}}(X,Y,A) &=_{def} \mathsf{Ability}(X, \texttt{"A"}, \mathsf{Claim}(X,Y,A)) \\ \overset{\leftarrow\to}{\mathsf{Power}}(X,Y,A) &=_{def} \mathsf{Ability}(X, \texttt{"A"}, \mathsf{Claim}(X,Y,\overline{A})). \end{aligned}$$

We will say that $\overset{\to\leftarrow}{\mathsf{Power}}$ represents *positive-force power* and that $\overset{\leftarrow\to}{\mathsf{Power}}$ represents *negative-force power*. As an example, suppose Bob and his daughter Alice go to the cinema for the first time and that he explains her that one ought not to talk loud in the cinema. If Bob is in a positive-power position towards Alice, she will comply with his request. If he is in a negative-power position, she will just do the opposite: the prohibition to talk loud would become for her an obligation to do so. The negative-force power position is neglected in the analytical literature, but it is critically important in institutional-construction terms: it posits the denial to recognize another normative system which attempts to positively enact a certain power. As a legal example, see for instance in the Dutch Declaration of Independence, the Act of Abjuration (1581) towards Spain [36].

From these concepts, we can define a new set of potestative relations $\mathsf{PR}^{\overleftarrow{\rightarrow}}$ as labels for a force-centered potestative square of opposition. More precisely, here the four formulas needed for the square are obtained by taking into account all possible combinations of positive- vs. negative-force power and Boolean negation:

$$\mathsf{PR}^{\overleftarrow{\rightarrow}} = \{\overset{\rightarrow\leftarrow}{\mathsf{Power}}(p,q,A), \overset{\leftarrow\rightarrow}{\mathsf{Power}}(p,q,A), \neg\overset{\leftarrow\rightarrow}{\mathsf{Power}}(p,q,A), \neg\overset{\rightarrow\leftarrow}{\mathsf{Power}}(p,q,A)\}.$$

Subalternation is here captured by the logical principle:

$$\forall x \forall y\ (\overset{\rightarrow\leftarrow}{\mathsf{Power}}(x,y,A) \rightarrow \neg\overset{\leftarrow\rightarrow}{\mathsf{Power}}(x,y,A)),$$

which is acceptable because otherwise the same stimulus `"A"` could generate two conflicting first-order relations.

4.4 *Outcome-centered potestative square of opposition*

So far, we have presented an abstract notion of power (expressing the ability of changing or the ability of not changing the target relation), and an operational notion (expressing the interaction between verbal directive and performance). We now proceed to the analysis of the 'canonic' form of power, captured by $\mathsf{Power}(p,q,A)$: the power of p to establish on q a duty to perform an action of type A. At a first glance one might be inclined to consider the statement $\mathsf{Power}(p,q,\overline{A})$ as contrary to the statement $\mathsf{Power}(p,q,A)$ (namely, the power to forbid A as contrary to the power to impose A), but this is not a valid choice: it may well be the case that p has the power to impose the duty to A, as well as to impose the prohibition to A.[12]

We focus then on the notion of *power to revoke a duty*. In this way, we can distinguish between a *positive-outcome* notion of power (i.e. the power to establish a duty) and a *negative-outcome* notion of power, that can be formally defined as:

$$\overline{\mathsf{Power}}(X,Y,A) =_{def} \exists\beta(\mathsf{Ability}(x,\beta,\neg\mathsf{Claim}(X,Y,A))).$$

As before, we can form a set of potestative relations PR as labels for (potentially) building an outcome-centered potestative square of opposition:

$$\mathsf{PR} = \{\mathsf{Power}(p,q,A), \overline{\mathsf{Power}}(p,q,A), \neg\overline{\mathsf{Power}}(p,q,A), \neg\mathsf{Power}(p,q,A)\}.$$

While looking whether subalternation is satisfied, we acknowledge the following logical principles, which provide conditions for the truth of statements involving outcome-centered notions of power:

$$\forall x \forall y\ (\mathsf{Power}(x,y,A) \rightarrow \neg\mathsf{Claim}(x,y,A))$$

$$\forall x \forall y\ (\overline{\mathsf{Power}}(x,y,A) \rightarrow \mathsf{Claim}(x,y,A)).$$

The rationale behind these principles is that power captures the potential of a manifestation, and so the manifestation must not hold, for the potentiality to hold. Thus, having a power to impose A on y entails that one does not already have a claim that A be performed by y. Analogously, having a power to license y from the performance of A entails that one has (until the power at issue will be exercised)

[12] This intuition can be confirmed by analyzing their truth-conditions: $\mathsf{Power}(p,q,A) \wedge \mathsf{Power}(p,q,\overline{A})$ entails (according to the principles that we will observe just below and the Law of Double Complementation) $\neg\mathsf{Claim}(p,q,A) \wedge \neg\mathsf{Claim}(p,q,\overline{A})$. Looking at the Claim-based deontic square of opposition in Figure 3, the latter is a conjunction of subcontrary statements, whence it can be true.

a claim that A be performed by y. These principles shed light on the way in which an Aristotelian square of opposition for outcome-centered modalities should be built.

According to the two logical principles given above, for the same normative party it is not possible to have the power to create a claim that A be performed by q *and* the power to revoke q from the duty of performing A. Thus, it is not possible that $\overline{\mathsf{Power}}(p,q,A)$ and $\mathsf{Power}(p,q,A)$ are true at the same time. Thus, this observation provides us with the subalternation principle

$$\forall x \forall y\ (\mathsf{Power}(x,y,A) \rightarrow \neg\overline{\mathsf{Power}}(x,y,A))$$

to construct a square of opposition by means of the set PR.

4.5 *Relationships between potestative squares*

After this elaboration, it is interesting to check which corners of squares of oppositions are occupied by the notions of power discussed thus far. Following the tradition (see e.g. [5]), we will make reference to the four corners in an Aristotelian square with the labels A (upper left corner), E (upper right corner), I (lower left corner) and O (lower right corner). Looking at Figure 3, one sees that $\mathsf{Power}^{\rightarrow\leftarrow}$ and Power occupy the A position in their respective squares, whereas $\mathsf{Power}^{\leftarrow\rightarrow}$ and $\overline{\mathsf{Power}}$ occupy the E position, Power^{+} occupies the I position and Power^{-} occupies the O position.

We can proceed further in the comparison between squares, in particular the potestative ones. The previous formulas can be applied to discover different relationships between the distinct forms of powers. First, the convergence or divergence with due performance in force-centered powers map directly or dually to outcome-centered powers:

$$\overset{\rightarrow\leftarrow}{\mathsf{Power}}(p,q,A) \rightarrow \mathsf{Power}(p,q,A) \qquad \overset{\leftarrow\rightarrow}{\mathsf{Power}}(p,q,A) \rightarrow \mathsf{Power}(p,q,\overline{A}).$$

Following the contrapositive, the absence of positive-outcome power to produce a duty (meaning that there is no triggering action to obtain this) maps *a fortiori* to the absence of a positive-force power for doing so; the same holds with respect to negative-outcome and negative-force power. These observations are captured by the implications below:

$$\neg\mathsf{Power}(p,q,A) \rightarrow \neg\overset{\rightarrow\leftarrow}{\mathsf{Power}}(p,q,A) \qquad \neg\mathsf{Power}(p,q,\overline{A}) \rightarrow \neg\overset{\leftarrow\rightarrow}{\mathsf{Power}}(p,q,A).$$

Second, positive-change power holds if any outcome-centered power holds:

$$\begin{aligned}\mathsf{Power}^{+}(p,q,A) \leftrightarrow \mathsf{Power}(p,q,A) &\vee \mathsf{Power}(p,q,\overline{A})\\ &\vee \overline{\mathsf{Power}}(p,q,A) \vee \overline{\mathsf{Power}}(p,q,\overline{A}).\end{aligned}$$

We have, dually, that:

$$\begin{aligned}\neg\mathsf{Power}^{+}(p,q,A) \leftrightarrow \neg\mathsf{Power}(p,q,A) &\wedge \neg\mathsf{Power}(p,q,\overline{A})\\ &\wedge \neg\overline{\mathsf{Power}}(p,q,A) \wedge \neg\overline{\mathsf{Power}}(p,q,\overline{A}).\end{aligned}$$

Introducing those relationships, we obtain the Aristotelian diagram in Figure 4.

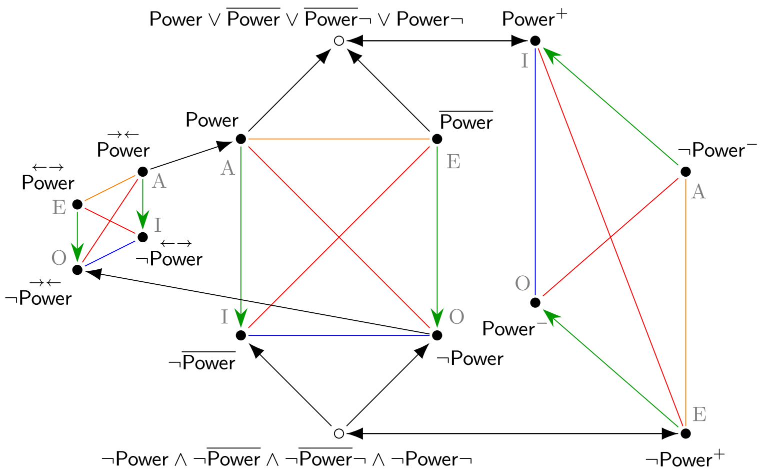

FIGURE 4. Map of potestative relations defined in terms of triggering action (force-centered square of opposition, the left one), in terms of outcome (middle square) and in terms of change or affecting outcomes (change-centered square of opposition, the right one). For visual clarity, labels of vertices are simplified so as to consist only of a (possibly negated) predicate without its arguments. Occurrences of negation before a predicate are standard, whereas occurrences after a predicate denote action complementation; for instance, we denote the power to issue a prohibition, i.e. $\mathsf{Power}(p,q,\overline{A})$, as $\mathsf{Power}\neg$. Notice that the leftmost square is vertically mirrored and the rightmost square underwent a 90 degree clockwise rotation. Colours are as usual except for the fact that examples of subalternation and logical equivalence between labels of different squares are in black, for better readability. Some Aristotelian relations between pairs of vertices of this diagram are omitted (e.g. the contradiction relation between the top and bottom vertices of the central hexagon).

5 Of lost symmetries

This section returns to the issue of the asymmetry observed between the analysis of first-order and second-order Hohfeldian diagrams. In particular, we observed that definitions of notions of the first square have an internal negation not present in the notions of the second square at the corresponding position:

$$\mathsf{Liberty}(X,Y,A) =_{def} \neg\mathsf{Claim}(Y,X,\overline{A})$$

$$\mathsf{Immunity}(X,Y,A) =_{def} \neg\mathsf{Power}(Y,X,A).$$

Our motivation to 'repair' the symmetry is to maintain the visual accessibility of the original Hohfeldian squares (Figure 1); in this sense, symmetry becomes a desired property not only among relations belonging to the same square (given that these should be all interdefinable, as in the Aristotelian squares), but also between pairs of relations belonging to the deontic and to the potestative square. Elaborations on these aspects will shed light on minimal informational requirements to specify normative relationships, distinguishing complete from incomplete positions.

5.1 Half-liberties and full-liberties, and disjoint or absolute duty

According to the description of the deontic square that we have provided in Figure 3, the formula $\mathsf{Liberty}(q,p,A)$ is logically equivalent to $\neg\mathsf{Claim}(p,q,\overline{A})$. However, this notion does not match the ordinary meaning of 'liberty', in the sense of being free to perform and to omit the action.[13] More precisely, the two statements $\neg\mathsf{Claim}(p,q,\overline{A})$ and $\neg\mathsf{Claim}(p,q,A)$—the latter logically equivalent to $\mathsf{Liberty}(q,p,\overline{A})$—are sub-contraries, which means that they cannot be both false. Yet, it may be that only one of $\mathsf{Liberty}(q,p,\overline{A})$ and $\mathsf{Liberty}(q,p,A)$ is true. Suppose, without loss of generalization, that $\mathsf{Liberty}(q,p,A)$ is true and that $\mathsf{Liberty}(q,p,\overline{A})$ is false: then, $\neg\mathsf{Liberty}(q,p,\overline{A})$ is true and so is $\mathsf{Claim}(p,q,A)$. The latter, in turn, is equivalent to $\mathsf{Duty}(q,p,A)$. Therefore, one gets that q is free to perform A with respect to p and, at the same time, has a duty towards p to perform A. In this case, speaking of a 'liberty' would be misleading. This is only a *half-liberty*, rather than a genuine one. By contrast, a *full-liberty* for q with respect to the performance of A obtains only when both formulas $\mathsf{Liberty}(q,p,A)$ and $\mathsf{Liberty}(q,p,\overline{A})$ are true as per the following definitional equivalence:

$$\mathsf{FullLiberty}(Y,X,A) =_{def} \mathsf{Liberty}(Y,X,A) \wedge \mathsf{Liberty}(Y,X,\overline{A}).$$

We can define correlatively a full no-claim relation:

$$\mathsf{FullNoClaim}(X,Y,A) =_{def} \neg\mathsf{Claim}(X,Y,A) \wedge \neg\mathsf{Claim}(X,Y,\overline{A}).$$

Now, being moved by the aim of an overall symmetry of the geometrical construction of deontic modalities, one could argue that there must be a notion of duty associated with the combination of the two formulas $\mathsf{Duty}(q,p,A)$ and $\mathsf{Duty}(q,p,\overline{A})$, which respectively correspond to $\mathsf{Claim}(p,q,A)$ and $\mathsf{Claim}(p,q,\overline{A})$ in the Claim-based deontic square. However, such a combination cannot be the joint truth of the two formulas. Indeed, if q were required both to perform A and to forbear from A, then there would be a conflict between norms, since q could not avoid doing something regarded as wrong. Here, we rather propose to further exploit the analogy between deontic and alethic modalities in order to find a more plausible solution. In fact, the notion of liberty is associated with the alethic notion of *possibility*; by contrast, the notion of claim and the correlative notion of duty are associated with the alethic notion of *necessity*. The square of opposition can thus be expanded to an *hexagon of opposition*, following the ideas in [6], in order to make room for two notions that respectively correspond with two-sided possibility and two-sided necessity. In the alethic case, the former notion is also known as *contingency*, the latter notion as non-contingency or *absoluteness* (see, e.g. [5], and [30].)

We can define, accordingly, an absolute duty, and a correlative absolute claim, confirming its duality with the full no-claim (see Figure 5):

$$\mathsf{AbsDuty}(Y,X,A) =_{def} \mathsf{Duty}(Y,X,A) \vee \mathsf{Duty}(Y,X,\overline{A})$$

$$\mathsf{AbsClaim}(X,Y,A) =_{def} \mathsf{Claim}(X,Y,A) \vee \mathsf{Claim}(X,Y,\overline{A})$$

$$\mathsf{FullNoClaim}(X,Y,A) =_{def} \neg\mathsf{AbsClaim}(X,Y,A)$$

[13] See, on similar lines, [26, 35].

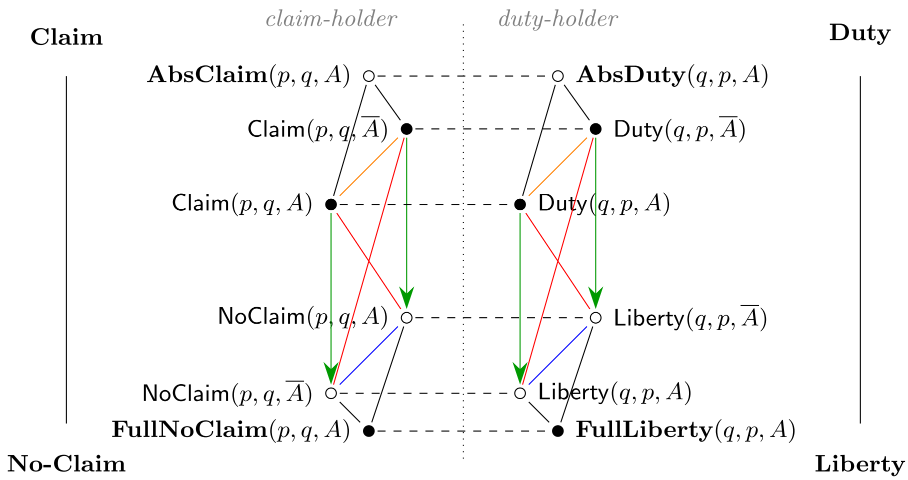

FIGURE 5. The two correlative hexagons of opposition for Hohfeld's deontic relations.

5.2 Full-disabilities, and absolute power

Similar considerations can be applied to the corresponding potestative concepts, resulting in the construction of full-disabilities, or absolute powers, and correlative notions. Figure 6 illustrates them for the outcome-centered power, but similar diagrams can be drawn for force-centered powers. In formulas:

$$\mathsf{AbsPower}(X,Y,A) =_{def} \mathsf{Power}(X,Y,A) \vee \overline{\mathsf{Power}}(X,Y,A)$$
$$\mathsf{FullDisability}(X,Y,A) =_{def} \neg\mathsf{AbsPower}(X,Y,A)$$
$$\mathsf{AbsLiability}(X,Y,A) =_{def} \mathsf{AbsPower}(Y,X,A)$$
$$\mathsf{FullImmunity}(X,Y,A) =_{def} \mathsf{FullDisability}(Y,X,A).$$

For change-centered powers, however, the definition would be different, as the contingency position would correspond to the presence of some actions that allow to affect the claim, and the presence of some actions that do not entail a modification of the claim.

5.3 Complete and incomplete positions

As observed amongst others in [7], positive duty (obligation), negative duty (prohibition) and full-liberty on performance (the three black dots in Figure 5) form a mutually exclusive *trichotomy* of normative concepts, providing sufficient meaningful information for agents to normatively qualify their conduct with respect to that performance. This is not the case for the set consisting of the two half-liberties and absolute duty (the white dots in Figure 5). For instance, suppose that the agent is aware of having an absolute duty; in order to know whether the duty is positive or negative, a query needs to be performed to identify whether a positive or negative half-liberty holds. Similar considerations apply w.r.t. positive power, negative power and full disability (for force-centered and outcome-centered powers). Knowing that a positive disability holds is not sufficient to entail that the agent does not have control; it may be simply that the claim has already been created.

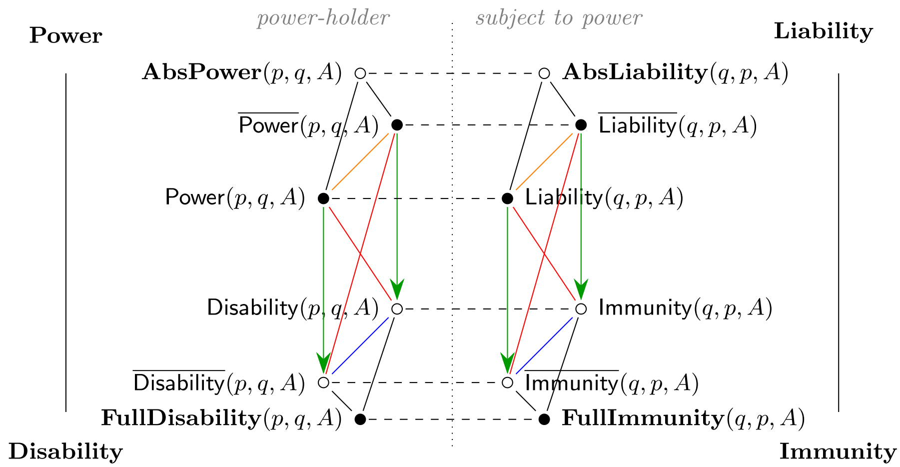

FIGURE 6. The two correlative hexagons of opposition for Hohfeld's potestative relations (outcome-centered).

6 Semantics

In the present section, we provide a model-theoretic semantics that can be used to interpret the formal language ($\mathcal{L}$) employed so far in our analysis of the Hohfeldian theory of normative relations. In the description of structures (Def. 6.1), we adopt the usual convention on symbols mentioned in section 3.1. For instance, p is an arbitrary element of $\texttt{Const}_{np}$, P is an arbitrary element of $\texttt{Pred} \setminus \{\mathsf{Ability}\}$, etc. Following this convention, we avoid quantifying over sets of symbols, except when ambiguity may arise.[14] The main novelties with respect to model-theoretic semantics for classical first-order logic is the use of typed domains of objects and the interpretation of the predicate $\mathsf{Ability}$.

DEFINITION 6.1 (Structure).
A structure to interpret language $\mathcal{L}$ is a tuple $\mathfrak{M} = \langle D_{\texttt{np}}, D_{\texttt{at}}, D_{\texttt{fm}}, I, G\rangle$ s.t.:

– $D_{\texttt{np}}$ is a domain of normative parties;
– $D_{\texttt{at}}$ is a domain of action types;
– $D_{\texttt{fm}}$ is the set of all formulas $\psi \in \mathcal{L}$, which have either the form $P(X, Y, A)$ or the form $\neg P(X, Y, A)$;
– I is an interpretation function s.t.:

  - $I(p) \in D_{\texttt{np}}$;
  - for $A \in \texttt{Const}^*_{at}$, $I(A) \in D_{\texttt{at}}$ and satisfies two conditions on identity, namely $I(A) = I(\overline{\overline{A}})$ and $I(A) \neq I(\overline{A})$[15] ;

[14] For instance, when using A, B, C etc., since sometimes we need to specify that these symbols denote only elements of $\texttt{Const}^*_{at}$, rather than elements of the larger set $\texttt{Term}^*_{at}$.

[15] We stress that the set $\texttt{Const}^*_{at}$ is denumerable by construction. When one assigns an interpretation to its members one can assume that it is a strict linear order (e.g. the order starts with the sequence of all atomic action types, then the sequence of all types of the form $\overline{a}$, then the sequence of all types of the form $\overline{\overline{a}}$, etc.), so as to get that $I(A)$ is determined before $I(\overline{A})$,

  - $I(P) \subseteq D_{\mathtt{np}} \times D_{\mathtt{np}} \times D_{\mathtt{at}}$;
  - $I(\mathsf{Ability}) \subseteq D_{\mathtt{np}} \times D_{\mathtt{at}} \times D_{\mathtt{fm}}$.
- G is the set of all assignment functions g s.t.:
  - $g(x) \in D_{\mathtt{np}}$;
  - $g(\alpha) \in D_{\mathtt{at}}$.

The sets $D_{\mathtt{np}}$, $D_{\mathtt{at}}$ and $D_{\mathtt{fm}}$ are taken to be non-empty and pairwise disjoint (the latter enjoys the property of non-emptiness as a consequence of Def. 3.1 and Def. 3.2). Moreover, the set D_{at} includes at least two elements, due to the requirement that, for each $A \in \mathtt{Const}^{*}_{at}$, $I(A) \neq I(\overline{A})$. When analyzing a structure $\mathfrak{M}$, we can write $D_{\mathtt{np}}(\mathfrak{M})$, $D_{\mathtt{at}}(\mathfrak{M})$, etc. to denote the elements of $\mathfrak{M}$ (as a tuple). The specification of $\mathfrak{M}$ can be omitted (hence, we can simply write $D_{\mathtt{np}}$, $D_{\mathtt{at}}$, etc.) when there is no risk of ambiguity.

Given $g, g' \in G(\mathfrak{M})$, g' is said to be a *V-variant* of g iff it differs from g at most on the value assigned to individual variable V. In order to simplify the presentation of truth-conditions, we will write ${}^{I}_{g}(X)$ to denote the *semantic value* of X, which corresponds to $I(X)$ if X is a constant and to $g(X)$ if X is a variable. The same holds for ${}^{I}_{g}(A)$.

DEFINITION 6.2 (Truth-conditions).
Formulas of $\mathcal{L}$ are evaluated with reference to a structure $\mathfrak{M}$ and an assignment $g \in G(\mathfrak{M})$, according to the truth-conditions below (where $\mathfrak{M}, g \vDash \phi$ means that ϕ is true with reference to $\mathfrak{M}$ and g, whereas $\mathfrak{M}, g \nvDash \phi$ means that ϕ is false with reference to $\mathfrak{M}$ and g):

- $\mathfrak{M}, g \vDash P(X, Y, A)$ iff $\langle {}^{I}_{g}(X),\ {}^{I}_{g}(Y),\ {}^{I}_{g}(A)\rangle \in I(P)$;
- $\mathfrak{M}, g \vDash \mathsf{Ability}(X, A, \psi)$ iff $\langle {}^{I}_{g}(X),\ {}^{I}_{g}(A),\ \psi\rangle \in I(\mathsf{Ability})$;
- $\mathfrak{M}, g \vDash X \asymp Y$ iff ${}^{I}_{g}(X) = {}^{I}_{g}(Y)$;
- $\mathfrak{M}, g \vDash A \asymp B$ iff ${}^{I}_{g}(A) = {}^{I}_{g}(B)$;
- $\mathfrak{M}, g \vDash \neg\phi$ iff $\mathfrak{M}, g \nvDash \phi$;
- $\mathfrak{M}, g \vDash \phi \rightarrow \psi$ iff $\mathfrak{M}, g \nvDash \phi$ or $\mathfrak{M}, g \vDash \psi$;
- $\mathfrak{M}, g \vDash \forall V \phi$ iff for every $g' \in G$, which is a V-variant of g, it holds that $\mathfrak{M}, g' \vDash \phi$.

DEFINITION 6.3 (Satisfiability and validity).
A formula $\phi \in \mathcal{L}$ is:

- *satisfiable in a structure* $\mathfrak{M}$ iff, for some $g \in G(\mathfrak{M})$, it holds that $\mathfrak{M}, g \vDash \phi$;
- *valid in a structure* $\mathfrak{M}$ iff, for every $g \in G(\mathfrak{M})$, it holds that $\mathfrak{M}, g \vDash \phi$;
- *satisfiable in a class of structures C* iff, for some structure $\mathfrak{M} \in C$, it holds that ϕ is satisfiable in $\mathfrak{M}$;
- *valid in a class of structures C* iff for every structure $\mathfrak{M} \in C$ it holds that ϕ is valid in $\mathfrak{M}$.

For reasons of brevity, here we just sketch how to obtain metalogical results with the semantics adopted and later illustrate how to use models in order to represent problems of normative reasoning. The notions of soundness and completeness of systems with respect to a class of structures are

$I(\overline{A})$ before $I(\overline{\overline{A}})$, etc. This procedure ensures that conditions for identity of semantic value are satisfied. For an analogous procedure in classical logic, see [4].

defined as usual (see, e.g. [4]). Our observations target systems containing the Law of Double Complementation, which are those originally analysed also in [28].

As far as the soundness of these systems is concerned, the crucial observation is the following. Given two action types $A, B \in \texttt{Const}^*_{at}$, it can be easily verified, via an induction on the syntactical complexity of formulas, that the schema $\phi \leftrightarrow \phi[A//B]$ is valid in structures satisfying the property $I(A) = I(B)$. Consequently, the Law of Double Complementation is valid in the class of all structures specified by Def. 6.1. From this, by adopting the procedure specified, e.g. in [4, 13], one can prove that if S is a logical system closed under the Law of Double Complementation and under all deductive principles of the Predicate Calculus with identity, then S is sound w.r.t. a subclass of all the structures specified by Def. 6.1.

As far as the completeness of systems containing the Law of Double Complementation is concerned, one can rely on an adaptation of the standard method of canonical models for theories of first-order logic [4, 13]. More precisely, suppose that one wants to check whether a closed formula ϕ is a consequence of a set of closed formulas Γ in a system S that is built over language $\mathcal{L}$ and contains the Law of Double Complementation; then, one can proceed as per the following steps, taking $\Gamma = \Gamma_0$ and $\mathcal{L} = \mathcal{L}_0$:

1. one builds a maximal S-consistent set Γ_0^{max} over $\mathcal{L}$, which contains Γ_0, relying on an enumeration of formulas; clearly, Γ_0^{max} will be closed under the Law of Double Complementation, since so is S (by assumption).
2. one moves to a language $\mathcal{L}_1 \supseteq \mathcal{L}$ and closes Γ_0^{max} under instantiation, namely adding a formula of the form $\phi[a/\alpha]$ for every formula $\exists\alpha\phi$ in Γ_1 and a formula of the form $\phi[p/x]$ for every formula $\exists x\phi$ in Γ^{max} (a and p being respectively an individual constant for action types and an individual constant for normative parties that belong to the vocabulary of $\mathcal{L}_1$ but not to the vocabulary of $\mathcal{L}$), thus getting a set Γ_1;
3. one performs step 1 with respect to Γ_1, thus getting a set Γ_1^{max};
4. for every Γ_n where $n \geq 1$, one performs first step 2 on Γ_n^{max}, thus getting Γ_{n+1}, and then step 1 on Γ_{n+1}, thus getting Γ_{n+1}^{max};
5. let $\Gamma_\infty = \bigcup_{n\in\mathbb{N}} \Gamma_n^{max}$, $\mathcal{L}_\infty$ be the language of Γ_∞, and $\texttt{Const}_{np\infty}$, $\texttt{Const}_{at\infty}$, etc. be the various relevant subsets of the vocabulary of Γ_∞ (following those mentioned in Def. 3.1); then one defines a structure $\mathfrak{M} = \langle D_{\texttt{np}}, D_{\texttt{at}}, D_{\texttt{fm}}, I, G\rangle$ where:
   - $D_{\texttt{np}} = \texttt{Const}_{np\infty}$;
   - $D_{\texttt{at}} = \texttt{Const}_{at\infty}$;
   - $D_{\texttt{fm}}$ is the set of all formulas $\psi \in \mathcal{L}_\infty$, which have either the form $P(X, Y, A)$ or the form $\neg P(X, Y, A)$;
   - the interpretation function I is defined as usual in Henkin's structures:
     - $\asymp$ is interpreted in such a way that $I(X) = I(Y)$ iff $X \asymp Y \in \Gamma_\infty$ and $I(A) = I(B)$ iff $A \asymp B \in \Gamma_\infty$[16] ;
     - every $P \in \texttt{Pred} \setminus \{\textsf{Ability}\}$ is interpreted as the class of triples (b, c, A) s.t. $b, c \in \texttt{Const}_{np}$, $A \in \texttt{Const}^*_{at}$ and $P(b, c, A)$ is a consequence of Γ_∞;
     - $\textsf{Ability}$ is interpreted as the class of triples (c, A, ψ) s.t. $c \in \texttt{Const}_{np}$, $A \in \texttt{Const}^*_{at}$ and $\textsf{Ability}(c, A, \psi)$ is a consequence of Γ_∞.[17]
   - G is the set of all possible assignments in $\mathfrak{M}$ (all those satisfying the conditions mentioned in Def. 6.1).

[16] Also in this case, one can rely on a strict linear order on $\texttt{Term}^*_{at}$ to determine the semantic value of all terms.

[17] Here and in the previous clause one could keep track of the additional 'stimulus' function mentioned in section 4.3 by replacing $A \in \texttt{Term}^*_{at}$ with $A \in \texttt{Term}^{*+}_{at}$.

Following the procedure in [13] —with a few adaptations due to the presence of typed sets of objects for normative parties ($D_{\texttt{np}}$) and action types ($D_{\texttt{at}}$), as well as of a set of formulas used as objects ($D_{\texttt{fm}}$)— one can prove that, for any $g \in G$, it holds that $\mathfrak{M}, g \models \psi$ whenever $\psi \in \Gamma$. Thus, $\mathfrak{M}$ is said to be the *canonical model* for Γ. The rest of the completeness argument runs as usual, by contraposition: if ϕ is not derivable from Γ in system S, then the above-sketched procedure illustrates how to build a canonical model for $\Gamma \cup \{\neg\phi\}$.

In the remaining part of this section, we provide an example of a model representing a normative problem. Consider the following scenario. Patricia (p) put her bike on sale and both Quentin (q) and Robert (r) made an offer. By officially accepting either of the two offers (a_1 or a_2, respectively), Patricia creates a claim on the corresponding bidder to buy her bike (b). This can be rendered as the following formula:

$$\mathsf{Ability}(p, a_1, \mathsf{Claim}(p, q, b)) \wedge \mathsf{Ability}(p, a_2, \mathsf{Claim}(p, r, b)).$$

For the sake of simplicity, we assume that we are working with a language vocabulary including only two predicates $\mathsf{Ability}$ and Claim, only three constants a_1, a_2 and b for action types and only three constants p, q and r for normative parties. Sets of variables can be arbitrarily chosen and sets of terms are obtained according to Def. 3.1. Let $\mathfrak{M}$ be a model such that:

– $D_{\texttt{np}} = \{\mathit{Patricia}, \mathit{Quentin}, \mathit{Robert}\}$;
– $D_{\texttt{at}} = \{\mathit{accepting}_1, \mathit{accepting}_2, \mathit{buying}\}$;
– $D_{\texttt{fm}} = \Gamma \ \cup \ \Gamma^{neg}$, where:

$$\Gamma = \{\mathsf{Claim}(X, Y, A) \mid X, Y \in \texttt{Term}_{np},\ A \in \texttt{Term}^*_{at}\}$$

and

$$\Gamma^{neg} = \{\neg\phi \mid \phi \in \Gamma\}$$

– $I(p) = \mathit{Patricia}$, $I(q) = \mathit{Quentin}$ and $I(r) = \mathit{Robert}$;
– $I(a_i) = \mathit{accepting}_i$ for $1 \leq i \leq 2$, and $I(b) = \mathit{buying}$;
– $I(\mathsf{Claim}) = \emptyset$ and $I(\mathsf{Ability}) = \{\langle \mathit{Patricia}, \mathit{accepting}_1, \mathsf{Claim}(p, q, b)\rangle, \langle \mathit{Patricia}, \mathit{accepting}_2, \mathsf{Claim}(p, r, b)\rangle\}$;
– G is the set of all assignment functions satisfying the conditions in Def. 6.1.

Due to the truth-conditions specified in Def. 6.2, the following holds for any $g \in G$:

$$\mathfrak{M}, g \models \mathsf{Ability}(p, a_1, \mathsf{Claim}(p, q, b)) \wedge \mathsf{Ability}(p, a_2, \mathsf{Claim}(p, r, b)).$$

Moreover, due to the definition of the predicate Power, the following holds:

$$\mathfrak{M}, g \models \forall x((x \not\approx p) \rightarrow (\neg\mathsf{Claim}(p, x, b) \wedge \mathsf{Power}(p, x, b))).$$

The last formal statement above provides a more detailed picture of the scenario: Patricia currently has no claim on either Quentin or Robert that they buy her bike; however, there are actions available to her, i.e. accepting either of the two offers, by means of which she can create a claim to buy on the corresponding bidder, so she is currently in a position of positive-outcome power with respect to both Quentin and Robert.

7 Implementation

This section presents relevant excerpts of the proposed conceptualization as an ASP (*answer set programming* [20]) program. The full code is publicly available.[18] In ASP, as in Prolog, the programmer models a problem in terms of rules and facts. The resulting code is given as input to a solver, which returns multiple answer sets or stable models that satisfy the problem. The main operational difference to Prolog is that all variables are grounded before performing search, and unlike SLDNF resolution, ASP solver algorithms always terminate. Although based on a different semantics, ASP programs share a good part of their syntax with Prolog; in principle, part of the proposed code can be used with small adjustments in Prolog, and then in computational agents embedding Prolog (e.g. in AgentSpeak-based languages).

Our implementation consists of two logic programming modules and various helper scripts to control and visualize the outcome of the solver. More in detail:

(a) an *atemporal model* defines all concepts starting from more fundamental primitives, following the definitions presented in the previous sections; it allows us to perform definitional queries in a given moment in time;
(b) a *temporal model*, consisting of Event Calculus axioms [19, 34], slightly modified to take into account the agents performing the actions, and dedicated axioms bridging normative to Event Calculus concepts.

For simplicity, rather than defining the transition functions over all notions, the temporal model is defined only on the most fundamental primitives. Therefore, in the current implementation,

(c) the helper controlling the reasoning cycle executes a two-step process:
- it *explores* the dynamics of primitives by executing with an interleaved semantics all actions which are relevant for transitions;
- it *expands* the description of this dynamics to other normative notions.

We will now present in detail the two models and provide an example of application.

7.1 Atemporal model

7.1.1 Hohfeldian modalities Following the previous theoretical elaboration, we consider the two predicates `claim/3` and `power/3` as primitives, and we then define the other Hohfeldian modalities from those. The first Hohfeldian square is expressed by the following rules ('`-`' denotes propositional negation):

```
noclaim(X,Y,A):- -claim(X,Y,A).
duty(Y,X,A):- claim(X,Y,A).
liberty(Y,X,A):- -claim(X,Y,neg(A)), action(A)
liberty(Y,X,neg(A)):- -claim(X,Y,A), action(A).
```

The definition of liberty has been encoded here with two rules. Implementing the law of double complementation for actions in an abstract form would have required to reify the normative primitives as a parameter, thus reducing readability, and to introduce axioms to prevent cycles, adding complexity. Instead, we have decided to specify where necessary rules in terms of *positive*

[18] https://github.com/gsileno/normative-primitives-asp

actions (selected by the condition `action(A)`), inflected in the two complementary cases. This operation is not necessary for the second Hohfeldian square, as there is no action complementation in the definitions:

```
disability(X,Y,A):- -power(X,Y,A).
liability(Y,X,A):- power(X,Y,A).
immunity(Y,X,A):- -power(X,Y,A).
```

These rules alone are however not sufficient for queries about normative relationships. First, because they were theoretically given as definitions, they need to be interpreted as equivalences, and therefore we need to add all inverse implications. Second, the reasoning method of ASP (and of Prolog) does not embed automatically the contrapositive of a given conditional in the inferential process. Each of the above rules has to map therefore to four rules, e.g.

```
noclaim(X,Y,A):- -claim(X,Y,A).
claim(X,Y,A):- -noclaim(X,Y,A).
-claim(X,Y,A):- noclaim(X,Y,A).
-noclaim(X,Y,A):- claim(X,Y,A).
```

In the following, when we will encode conditionals we will omit to report the contrapositive rules, and, in case of definitions, of the inverse ones.

7.1.2 Aristotelian diagrams To model the deontic square of opposition, we need to add the relation of subalternation

```
-claim(X,Y,neg(A)):- claim(X,Y,A).
```

For the potestative concepts, we need only to provide the definitions, as subalternation relations can be proven to be properties of their internal dynamics. We define positive and negative outcome powers directly from ability, which is the concept with the finest granularity:

```
power(X,Y,A):- ability(X,B,plus(claim(X,Y,A))).
negpower(X,Y,A):- ability(X,B,minus(claim(X,Y,A))).
```

Note that the predicate 'ability' is used here differently than in the theoretical section. Instead of asserting the negation of a claim, negative power retracts a positive claim. This choice is preparatory to the subsequent temporal model, based on Event Calculus, for which only positive fluents are kept. We use ability also to define positive- and negative-force powers:

```
posforce:power(X,Y,A):-
 ability(X,require(A),plus(claim(X,Y,A))).
negforce:power(X,Y,A):-
 ability(X,require(A),plus(claim(X,Y,neg(A)))).
```

For the change-centered potestative square, we need first to introduce $\mathsf{Power}_{OReilly}$:

```
power_oreilly(X,Y,B,A):- ability(X,B,plus(claim(X,Y,A))).
power_oreilly(X,Y,B,A):- ability(X,B,plus(claim(X,Y,neg(A)))).
power_oreilly(X,Y,B,A):- ability(X,B,minus(claim(X,Y,A))).
power_oreilly(X,Y,B,A):- ability(X,B,minus(claim(X,Y,neg(A)))).
```

And from this we define the positive change and no-change powers:

```
poschange_power(X,Y,A):- power_oreilly(X,Y,B,A).
nochange_power(X,Y,A):- -poschange_power(X,Y,A).
```

7.1.3 Full positions The definition of full positions (the ones which are most relevant for reasoning purposes) is straightforward:

```
full_liberty(X,Y,A):-
 liberty(X,Y,A), liberty(X,Y,neg(A)), action(A).
full_disability(X,Y,A):-
 disability(X,Y,A), disability(X,Y,neg(A)), action(A).
```

7.1.4 Normative closures To make the normative system complete, the reasoner should always be able to conclude upon a certain result. To deal with ignorance, we need to introduce defaults, and this can be done by means of default negation ('`not`' in ASP/Prolog) used to reify a propositional negation. Because we have two fundamental primitives (`claim` and `ability`), we introduce two closures as illustrative examples. For the deontic square, if not prohibited/requested explicitly, an agent is not prohibited/requested (also known as *weak permission*):

```
-claim(X, Y, neg(A)):- not claim(X, Y, neg(A)),
 agent(X), agent(Y), action(A), X != Y.
```

For the potestative square, if not empowered explicitly, the agent does not have that power. We specify this closure in terms of ability:

```
-ability(X, require(A), plus(claim(X, Y, A))):-
 not ability(X, require(A), plus(claim(X, Y, A))),
 agent(X), agent(Y), action(A), X != Y.
```

The source code includes complementary rules.

7.2 Temporal model

To apply normative directives in real settings, we cannot avoid embedding these notions in time. Efficient temporal reasoning is an area of research in itself, outside the scope of the present paper; it is however still relevant to provide an example of how the previous model can be integrated in an *off-the-shelf* engine for temporal reasoning, also to identify and discuss the potential challenges.

7.2.1 Event calculus We will base our implementation on Event Calculus [19, 34], a well-known solution for temporal reasoning in logic programming (for an efficient variant see e.g. [3]). Event Calculus distinguishes *fluents*, i.e. dynamic properties of the world, from *events*, transitions that trigger a change of fluents. In our domain, because all normative relationships may change they can be seen as potential fluents. The events that modify these fluents are actions performed by agents. Taking these observations into accounts, we slightly modify the Event Calculus axioms as:

```
holds(F,T):-
 holds(F,0), not clipped(0,F,T), fluent(F), time(T).
holds(F,T2):- does(X,A,T1), initiates(X,A,F,T1), T1<T2,
```

```
 not clipped(T1,F,T2), fluent(F), time(T1), time(T2).
clipped(T1,F,T2):- does(X,A,T), T1<=T, T<T2,
 terminates(X,A,F,T), time(T1), time(T2).
```

where we replace predicates as `occurs(E,T)` with `does(X,A,T)`, and similarly `initiates(E,F,T)` with `initiates(X,A,F,T)`.

7.2.2 Dynamics of normative concepts Based on this machinery, we now translate the dynamics of normative concepts. The prototypical normative concept associated with change is `ability`, and correspondingly `power`; actually, the predicate `ability` can be seen [37] as reifying the presence of a causal mechanism, and be put in direct correspondence with the `initiates` predicate in Event Calculus:

```
initiates(X,A,F,T):- holds(ability(X,A,plus(F)),T).
```

Enacting a power means also consuming it, as the consequence will already exist, and change cannot be produced again. Consumption can be specified via the `terminates` predicate:

```
terminates(X,A,ability(X,A,plus(F)),T):-
 holds(ability(X,A,plus(F)),T).
```

The presence of a prohibition with respect to an action (or negative duty) makes the performance of the action a violation of that prohibition; complementarily, the presence of a (positive) duty makes non-performance a violation. These *count-as* rules can be written as:

```
does(Y,violate(claim(X,Y,A)),T):-
 does(Y,neg(A),T), holds(claim(X,Y,A),T), action(A).
does(Y,violate(claim(X,Y,neg(A))),T):-
 does(Y,A,T), holds(claim(X,Y,neg(A)),T), action(A).
```

In our model, we specify *violate-directive* actions to produce *directive-violated* fluents. Similar rules are encoded for fulfillment, and for the removal of a duty after fulfillment. A delicate part for using the second of the above rules lies in specifying *negative actions*. As a practical solution, we utilize default negation in relevant conditions:

```
does(X,neg(A),T):- holds(claim(_,X,A),T),
 not does(X,A,T), agent(X), action(A), time(T).
```

i.e. agents perform a negative action (to be interpreted as to *omit* an action) if performance is due, and they do not perform.

7.3 Use case: normative exploration

The models above can be used for various normative reasoning applications. Along with this work, we implemented an application for contractual exploration, to exploit the optimized search capabilities of ASP. The exploration is lead by the following disjoint rules, specifying *whenever an action is normatively relevant, it may be performed*.

```
{does(X,A,T)}:- holds(ability(X,A,_),T), action(A).
{does(Y,A,T)}:- holds(claim(_,Y,A),T), action(A).
{does(Y,A,T)}:- holds(claim(_,Y,neg(A)),T), action(A).
```

We consider two further constraints to avoid inconsistent and irrelevant execution paths. *Only one positive action per time step* (interleaved semantics):

```
:- does(X1,A1,T), does(X2,A2,T),
 X1!=X2, A1!=A2, action(A1), action(A2).
```

Inaction is never followed by action (equivalently, inaction is a terminal action):

```
some_action_occurs(T):- does(_,A,T).
:- not some_action_occurs(T), some_action_occurs(T+1), time(T).
```

The dynamics of normative concepts defined in the temporal model is specified only around the notions of ability and of claim. The exploration suggested above will therefore provide the dynamics concerning these two fundamental concepts. How to connect this result with all the other notions? Rather than rewriting all dynamics accordingly, we developed a two-step process: (i) the temporal model provides the fundamental backbone of the dynamics, and then (ii) the atemporal model is applied at each time-step of the output of the temporal model, to enlarge the description to other concepts. Note that this is only an example of utilization of the given models. If adequately converted to Prolog programs (e.g. by adding a predicate for classic negation), they could be used for normative querying without materializing all facts as in the ASP implementation.

Example of application: a bilateral sale Consider the specification of a potential sale bilateral contract. In words, the target normative construct entails that John, by offering, provides Paul the power to accept, by which in turn he creates the duty upon John to deliver, and upon himself to pay. We encode this in ASP by using the `ability` predicate (which in our framework is the finest granularity for potestative concepts):

```
agent(john). agent(paul).
action(offer). action(accept). action(pay). action(deliver).
holds(ability(john,offer,plus(
 ability(paul,accept,plus(claim(john,paul,pay)))
)),0).
holds(ability(john,offer,plus(
 ability(paul,accept,plus(claim(paul,john,deliver)))
)),0).
time(0..3).
```

Feeding this model to the exploration method introduced above, we obtain 9 execution paths possible across 3 time steps with the given execution semantics:

- Nothing happens (i.e. John does not offer).
- John offers.
- John offers; Paul accepts; Paul does not pay, John does not deliver (they both violate their duties).
- John offers; Paul accepts; Paul pays (fulfills his duty) but John does not deliver (violates his duty); John still does not deliver.
- John offers; Paul accepts; Paul pays (fulfills his duty) but John does not deliver (violates his duty); John delivers (fulfills his duty, but was already violated).

– John offers; Paul accepts; Paul does not pay, John does not deliver; Paul pays, John does not deliver.
– John offers; Paul accepts; Paul does not pay, John does not deliver; Paul does not pay, John delivers.
– John offers; Paul accepts; Paul does not pay, John delivers; Paul still does not pay.
– John offers; Paul accepts; Paul does not pay, John delivers; Paul pays.

Because we have selected an interleaved semantics (only one action per time step), there is always some violation after acceptance, as the action of accepting generates two concurrent duties. For a more sound model, one may relax this constraint, or also introduce a temporal dimension in the specification on duties (see e.g. [11]).

By using the helpers we developed, descriptions of normative relationships are generated at each time step in each execution paths. For instance, for the scenario in which a deal is made, Paul pays but John does not deliver (4th case in the list above), we have:

```
.. time 0
=> does(john, offer)
.. time 1
paul upon john: power to require deliver
john towards paul: liability to be required to deliver
=> does(paul, accept)
.. time 2
john towards paul: claim to pay
john towards paul: duty to deliver
paul towards john: claim to deliver
paul towards john: duty to pay
=> does(paul, pay),
 (does(john, neg(deliver)),
 (does(paul, fulfill(claim(john, paul, pay))),
 (does(john, violate(claim(paul, john, deliver)))
.. time 3
** VIOLATIONS ** claim(paul, john, deliver)
paul towards john: claim to deliver
john towards paul: duty to deliver
=> does(john, neg(deliver)),
 (does(john, violate(claim(paul, john, deliver))).
```

Note how the helper at the moment extracts only the canonic form of power (for the acceptance), in alignment with the most common standpoint in legal scholarship, but this is not sufficient to express all normative mechanisms in place.

8 Final remarks

In our effort of systematizing the analysis of Hohfeld's theory of normative relations, we have acknowledged that an expressive formal language and diagrams of opposition are effective tools to discover gaps in the theoretical framework and to deepen some conceptual issues (such as various senses of power defined in terms of ability). Furthermore, the choice of Aristotelian diagrams paves the way to using our framework in the development of dedicated computer-assisted methods for normative reasoning. As a matter of fact, these diagrams may be employed to create user-friendly interfaces for exploring legal documents such as contracts: rather than inspecting hundreds of sentences in the text of a contract, a subject may more easily figure out her normative relations (duties, rights, etc.) with the other parties by navigating a diagrammatic model of the contract. Due to the cognitive efficacy of visual aids, diagrams are potentially accessible to a broader audience than other logic-based methods for assisted normative reasoning. Moreover, the wide presence in human conceptualizations of polygons of opposition as knowledge constructs may be due to special properties for derivation/reasoning. We are indeed exploring this direction and the initial results are promising, with a syntactic method that allows one to derive a finite set of normative relations from another in polynomial time [29].

At the moment, the greatest gap in this line of work lies is the following problem: how to map the (causal?) connection between power and duty within the derivation mechanism. The implementation part of this paper was indeed primarily meant to make clear to the reader some of the challenges involved in *applying* normative concepts on cases unfolded in time. Beside the use of specific axioms for the dynamics (here in the form of Event Calculus), we needed to add default negation in order to make the normative system complete, and more importantly, to give an operational meaning to normative concepts, i.e. to specify the role(s) they play in the dynamics of the normative system (eg. violations, fulfillments,..). Our future goals include to study whether such dynamics can be traced back to some diagrams, and whether there are other atemporal diagrammatic structures that remain to be unveiled. The study of an extension to the formal language $\mathcal{L}$ to include nested forms of ‘ability’ can be seen as one of the theoretical steps in support of such direction.

A second dimension we aim to explore further is the passage from individual normative parties to roles. In legal scholarship, Hohfeld’s framework is usually deemed to be suitable for a *subjective right* stance (focus on individuals), but to be problematic with respect to *objective rights* (eg. *erga omnes*). Following the most common approach, *erga omnes* directives could be translated as directives towards each individual of certain sets (see e.g. [32]). We are however studying also an alternative approach, which would give roles the status of first-class citizens in the formalism, and allow individuals to embody different roles. Despite being more fitting to the form of law (legal norms are never about specific individuals), this approach comes with additional complexity, as roles may be conflicting, thus requiring machinery to deal with (role) defeasibility.

Acknowledgements

Matteo Pascucci was supported by the *Štefan Schwarz Fund* for the project ‘A fine-grained analysis of Hohfeldian concepts’ (2020–2023) and by the VEGA grant no. 2/0125/22 ‘Responsibility and modal logic’. Giovanni Sileno was partly funded by the Dutch Research Council (NWO) for the HUMAINER AI project (KIVI.2019.006). The article results from a joint research work of the two authors. Sections 1–5 are equally produced by both authors, section 6 is due to M.P. and section 7 is due to G.S.

Conflict of interest

The present work does not give rise to any conflict of interest.

References

[1] C. E. Alchourrón. Logic of norms and logic of normative propositions. *Logique et Analyse*, **12**, 242–268, 1969.

[2] A. R. Anderson. The logic of Hohfeldian propositions. *Logique et Analyse*, **13**, 231–244, 1970.

[3] A. Artikis, M. Sergot and G. Paliouras. An event calculus for event recognition. *IEEE Transactions on Knowledge & Data Engineering*, **27**, 895–908, 2015.

[4] J. Barwise. An introduction to first-order logic. In *Handbook of Mathematical Logic*, 8th edn., J. Barwise., ed., pp. 5–46, 1993.

[5] J.-Y. Béziau. The power of the hexagon. *Logica Universalis*, **6**, 1–43, 2012.

[6] R. Blanché. Sur l'opposition des concepts. *Theoria*, **19**, 89–130, 1953.

[7] J. A. de Oliveira Lima, C. Griffo, J. P. A. Almeida, G. Guizzardi and M. I. Aranha. Casting the light of the theory of opposition onto Hohfeld's fundamental legal concepts. *Legal Theory*, **27**, 2–35, 2021.

[8] M. De Vos, S. Kirrane, J. Padget and K. Satoh. Odrl policy modelling and compliance checking. In *Rules and Reasoning*, P. Fodor, M. Montali, D. Calvanese and D. Roman., eds, pp. 36–51. Springer International Publishing, Cham, 2019.

[9] L. Demey. Computing the maximal Boolean complexity of families of Aristotelian diagrams. *Journal of Logic and Computation*, **28**, 1323–1339, 2018.

[10] A. Dickey. Hohfeld's debt to Salmond. *University of Western Australia Law Review*, **59**, 1971.

[11] F. Dignum, H. Weigand and E. Verharen. Meeting the deadline: on the formal specification of temporal deontic constraints. *Foundations of Intelligent Systems, LNAI*, **1079**, 243–252, 1996.

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

[13] L. Henkin. The completeness of the first-order functional calculus. *The Journal of Symbolic Logic*, **14**, 159–166, 1949.

[14] W. N. Hohfeld. Some fundamental legal conceptions as applied in judicial reasoning. *Yale Law Journal*, **23**, 16–59, 1913.

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

[16] A. J. I. Jones and M. Sergot. A formal characterisation of institutionalised power. *Journal of IGPL*, **4**, 427–443, 1996.

[17] S. Kanger and H. Kanger. Rights and parliamentarism. *Theoria*, **32**, 85–115, 1966.

[18] A. Kocourek. The Hohfeld system of fundamental legal concepts. *Illinois Law Review*, **15**, 24–39, 1920.

[19] R. Kowalski and M. Sergot. A logic based calculus of events. *New Generation Computing*, **4**, 67–95, 1986.

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

[21] L. Lindahl. Position and change: a study in law and logic. In *Synthese Library*. Springer, 1977.

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

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

[24] T. Novotná and M. Pascucci. Normative parties in subject position and in object position. In *The Logica Yearbook 2020*, M. Blicha and I. Sedlár., eds, pp. 147–164, 2021.

[25] C. Nyquist. Teaching Wesley Hohfeld's theory of legal relations. *J. Legal Educ.*, **52**, 238, 2002.

[26] D. T. O'Reilly. Using the square of opposition to illustrate the deontic and alethic relations. *University of Toronto Law Journal*, **45**, 279–310, 1995.

[27] J. Padget, E. E. D. Elakehal, T. Li and M. De Vos. Instal: an institutional action language. In *Social Coordination Frameworks for Social Technical Systems*, pp. 101–124. Springer, 2016.

[28] M. Pascucci and G. Sileno. The search for symmetry in Hohfeldian modalities. In *Diagrammatic Representation and Inference. Proceedings of Diagrams 2021*, A. Basu, G. Stapleton, S. Linker, C. Legg, E. Manalo and P. Viana., eds, pp. 87–102. Springer, 2021.

[29] M. Pascucci and G. Sileno. Computability of diagrammatic theories for normative positions. In *Proceedings of JURIX 2021: International Conference on Legal Knowledge and Information Systems*, E. Schweighofer., ed., pp. 171–180, 2021.

[30] C. Pizzi. Contingency logics and modal squares of opposition. In *New Dimensions of the Square of Opposition*, J. Y. Béziau and K. Gam-Krywoszynska., eds, pp. 201–222, 2012.

[31] J. W. Salmond. Jurisprudence or the theory of the law. *Alpha Editions*, 1902, 2019.

[32] G. Sartor. Fundamental legal concepts: a formal and teleological characterisation. *Artificial Intelligence and Law*, **14**, 101–142, 2006.

[33] M. Sergot. A computational theory of normative positions. *ACM Transactions on Computational Logic (TOCL)*, **2**, 581–622, 2001.

[34] M. Shanahan. The event calculus explained. *Artificial Intelligence Today*, 409–430, 1999.

[35] G. Sileno. *Aligning Law and Action*. PhD Thesis, University of Amsterdam, 2016.

[36] G. Sileno, A. Boer and T. van Engers. On the interactional meaning of fundamental legal concepts. In *Proceedings of the 27th International Conference on Legal Knowledge and Information Systems (JURIX 2014), volume FAIA 271*, R. Hoekstra., ed., pp. 39–48, 2014.

[37] G. Sileno, A. Boer and T. van Engers. Towards a computational theory of action, causation and power for normative reasoning. In *Proceedings of the 32th International Conference on Legal Knowledge and Information Systems (JURIX 2019)*, pp. 211–216, 2019.

[38] G. Sileno and M. Pascucci. Disentangling deontic positions and abilities: a modal analysis. In *Proceedings of the 35th Edition of the Italian Conference on Computational Logic (CILC 2020), volume 2710*, F. Calimeri, S. Perri and E. Zumpano., eds, pp. 36–50, 2020.

[39] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, vol. **13**. MIT Press, 2000.

[40] L. W. Sumner. *The Moral Foundation of Rights*. Clarendon Press, 1987.

Received 8 June 2022