Chapter 1
Deontic Logics based on Boolean Algebra

Pablo F. Castro and Piotr Kulicki

Abstract Deontic logic is devoted to the study of logical properties of normative predicates such as *permission*, *obligation* and *prohibition*. Since it is usual to apply these predicates to actions, many deontic logicians have proposed formalisms where actions and action combinators are present. Some standard action combinators are action conjunction, choice between actions and *not doing* a given action. These combinators resemble boolean operators, and therefore the theory of boolean algebra offers a well-known mathematical framework to study the properties of the classic deontic operators when applied to actions. In his seminal work, Segerberg uses constructions coming from boolean algebras to formalize the usual deontic notions. Segerberg's work provided the initial step to understand logical properties of deontic operators when they are applied to actions. In the last years, other authors have proposed related logics. In this chapter we introduce Segerberg's work, study related formalisms and investigate further challenges in this area.

1.1 Introduction

The so-called *boolean operators* (*or*, *and*, *not*) are commonly used in ordinary language as basic connectors in phrases to put together propositions, subjects and verbs. George Boole in his famous text *An Investigation of the Laws of Thought* [5] was one of the first mathematicians (if not the first) to study the mathematical properties of these connectors, his work is considered a cornerstone of modern logic, and can be thought of as capturing some universal laws of logic. One of the main contributions

Pablo F. Castro
Universidad Nacional de Río Cuarto, Ruta Nacional No. 36, Km. 601, Río Cuarto, Cordoba, Argentina, e-mail: pcastro@dc.exa.unrc.edu.ar

Piotr Kulicki
John Paul II Catholic University of Lublin, Al. Raclawickie 14, 20-950 Lublin, Poland e-mail: kulicki@l3g.pl

of George Boole to logic was the characterization of logical reasoning by means of algebraic equations. Since then, boolean algebra and its generalizations (boolean algebras with operators [15, 16]) have been used to study the mathematical properties of logics by means of algebras. A boolean algebra is made up of a non-empty set of elements, binary operators $+, \times$, the unary operator $-$ and two distinguished constants 0 and 1. Several (complete) axiomatizations of boolean algebras have been proposed in the literature; the following axiomatization comes from [12].

- $-0 = 1$ **and** $0 = -1$ *(Zero and One laws).*
- $x \times 0 = 0$ **and** $x + 1 = 1$ *(Absorption of zero and one laws).*
- $x \times 1 = x$ **and** $x + 0 = x$ *(Identity laws).*
- $x \times -x = 0$ **and** $x + -x = 1$ *(Inverse laws).*
- $-(-x) = x$ *(Involution law).*
- $x \times x = x$ **and** $x + x = x$ *(Idempotent laws).*
- $-(x \times y) = -x + -y$ **and** $-(x + y) = -x \times -y$ *(De Morgan laws).*
- $x \times y = y \times x$ **and** $x + y = y + x$ *(Commutativity laws).*
- $x \times (y \times z) = (x \times y) \times z$ **and** $x + (y + z) = (x + y) + z$ *(Associativity laws).*
- $x \times (y + z) = (x \times y) + (x \times z)$ **and** $x + (y \times z) = (x + y) \times (x + z)$ *(Distributivity laws).*

This set of axioms is not the smallest one possible, but it exposes the standard properties of boolean algebras. It is straightforward to see that these properties are true for set intersection, set union and set complement in any field of sets. One may think of logical propositions such as *it is raining* or *the wall is white* as elements of a boolean algebra; and therefore the boolean operators allow us to construct more complicate statements, such as: *it is raining or it is sunny*; *the wall is not white*; *it is raining and the wall is white*. As a consequence, propositional logic can be seen as a boolean algebra, the formal technique to connect both worlds is called Lindenbaum-Tarski algebra, which is a boolean algebra made up of equivalence classes of sentences and the corresponding operations [29].

Two useful concepts that we will use through this chapter are those of **ideal** and **filter**; an ideal I of a boolean algebra B is a non-empty set $I \subseteq B$ satisfying the following conditions:

1. If $x \in I$ and $y \in I$, then $x + y \in I$,
2. If $x \in I$ and $y \in B$, then $x \times y \in I$.

The dual notion of ideal is called filter: a filter is a non-empty subset $F \subseteq B$ such that it satisfies:

1. If $x \in F$ and $y \in F$, then $x \times y \in F$,
2. If $x \in F$ and $y \in B$, then $x + y \in F$.

An ideal that is not a (proper) subset of another ideal is called *maximal ideal*; on the other hand, maximal filters are called *ultrafilters*; and they are one of the key notions of boolean algebra, for instance, *ultrafilters* are usually used for proving Stone's representation theorem [29]. We do not intend to introduce boolean algebras in detail in this chapter, good references are [29, 12].

Let us take another possible intuitive view of a boolean algebra: we may think of actions as elements of a boolean algebra, and so action combinators are the operations in this algebra. For instance, one may think of the action of driving as the set of all the ways in which one may drive: *driving fast*, *driving slow*, etc. Let us note that the boolean operators capture the way in which these sets can be combined; for example, consider the action of *driving* and the action of *drinking*, the boolean operators allow us to consider the following actions: *driving or drinking*; *driving and drinking*; *not driving*, etc. Roughly speaking, the first action expresses a choice between actions: one may perform any of these actions; the second one expresses an execution of two actions at the same time: one is driving and drinking; while the third one captures the notion of alternative action: one performs an action other then driving.

That is, at first sight, boolean algebras provide a useful mathematical framework to study basic properties of actions when they are combined in a simple way. In that framework different properties of actions can be analyzed. One type of such properties is the normative value of actions, which is investigated within deontic logic. Deontic logic can be most generally defined as a logic for rational agents acting in situations in which some kind of norms regulating their behaviour is present. The norms can be of a various nature - moral, legal, technical, organizational. Deontic action logic is a branch of this discipline in which norms are applied to actions (alternatively norms might be linked with states of affairs).

Within deontic action logic, the deontic value of boolean combinations of basic actions is worthy of being investigated. For example, if the action of drinking is permitted to be performed in any scenario (that is, it is allowed in a strong sense), then it is natural to think that we are allowed to drink while performing any other action (e.g., drinking while driving); in the interpretation of actions given above, this implies that permitted actions form an ideal in the algebra of actions. We discuss these ideas in detail in section 1.3.

Let us remark that deontic logic is naturally related to the study of the logical properties of actions; St. Anselm, who investigated the properties of the Latin expressions *facere* and *non facere*, is considered the precursor of the formal study of actions and related concepts; his work has been an inspiration for contemporary authors, the reader can find a detailed introduction to the history of logic of actions in [28]. Modern logic of actions starts with the works of Belnap (*stit* logic) [3], Kanger [19], von Wright [34] and Segerberg [27] between others. In this text, we focus on those works where boolean algebras are used as a formalism to capture the properties of actions when combined with deontic predicates.

The chapter is organized as follows. In the next section we briefly review the history of deontic logic before Segerberg's work. In section 1.3 we introduce Segerberg formalism with some remarks; in section 1.4 we introduce review some contemporary works in deontic logic based on boolean algebra. Finally, in the last sections, we investigate future lines of research, and present some final remarks.

1.2 Deontic action logic before Segerberg

Elements of logic of norms, preferences and imperatives were present all along the history of logic. First traces of formalization of deontic reasonings can be found already in the works of Aristotle, Aquinas and G.W. Leibniz. In modern times it was followed by the works of authors (philosophers, logicians and theorists of law) such as B. Bolzano, A. Hofler, E. Husserl, G.E. Moore, E Westermarck, P. Lapie, E. Mally, K. Menger, W. Dubislav, J. Jorgensen, A. Ross, A. Hofstadter, J.C.C. McKinsey, R.M.Hare, R. Rand, but these works lack formal development or clarity in the understanding of the nature of norms. Thus, they cannot be treated as mature logical systems. We shall not present the details of those works, one can find a detailed presentation in [18].

The beginning of contemporary deontic logic is connected with von Wright's work published in 1951 [33], in which he presented the first system of that kind with the use of techniques of formal logic as we understand it by now.[1]

There are two main assumptions of this system. Firstly, deontic notions (from which von Wright is interested in obligation, permission and forbiddance) are applied to actions. Secondly, deontic notions are treated as modal operators along with alethic, epistemic and existential modalities. Thus, obligatory is understood as an analogous of (alethic) necessary, (epistemic) known and (existential) for all, permitted – possible, undecided and for some, and finally forbidden – impossible, falsified and for some but not for all.

After von Wright's first paper, most of the work in deontic logic followed the second assumption neglecting the first one. What was created then is usually called standard deontic logic formally built in the same way as other modal systems, in which propositions are arguments of modal operators. It was Segerberg who reversed this tendency.

Von Wright, already in his first paper, points out a few more important issues. He distinguishes types of actions from individual actions. He calls the first ones acts, and understands them as properties of individual actions defining a type and act-individuals - particular actions. In his system he uses the first ones. He assumes that there is a finite number of atomic acts from which one can create complex acts using boolean operators. He called such complex actions *molecular complexes*. The same symbols were used for the operators for creating complex acts as well as for truth functions. That made it easy to shift to standard deontic logic. However, at that stage they were intuitively divided and consequently the nesting of deontic operators was not possible.

Von Wright did not introduce any formal semantics for his first deontic system. Instead, he formed several laws of deontic logic which he used as a foundation of his system. They were described as follows.

- A Principle of Deontic Distribution

[1] von Wright in [35] lists three 'founding fathers' of modern deontic logic: himself, J. Kalinowski and O. Becker. All of them published their first papers on deontic logic in early 1950s we shall concentrate on the work of von Wright is closest to.

“If an act is a disjunction of two other acts, then the proposition that the disjunction is permitted is the disjunction of the proposition that the first act is permitted and the proposition that the second act is permitted” ([33] page 7).
Let us remark that an analogous principle for conjunction does not hold.

- A Principle of Permission
“Any given act is either itself permitted or its negation is permitted” ([33] page 9).
- A Principle od Deontic Contingency
“A tautologous act [an act that is performed no matter what an agent does] is not necessarily obligatory, and a contradictory act is not necessarily forbidden” ([33] page 11).

Since nesting of deontic operators is not allowed, the Principle of Deontic Distribution and the Boolean character of operators on acts imply that every deontic proposition can be transformed to a form called by von Wright *absolutely perfect disjunctive normal form*. This normal form can be used for the verification of deontic propositions.

Some other contributions to deontic logic of action logic, which occurred between the first works of von Wright on deontic logic and Segerberg’s works, are also worth mentioning. The first of them is a strict distinction between names of actions and propositions introduced in [17]. That was related to a division of the field to deontic logic of action (ought-to-do logic) and deontic logic of states (ought-to-be logic).

Another important contribution was the introduction of formal semantics into deontic action logic. It took the form of 3-valued matrices. In [17] a matrix for negation was presented and in [10] the idea was extended to conjunction and disjunction of actions (being the concept of action or ‘inner’ counterparts of operators of the propositional calculus). Aquist in [2] has shown that using matrices results in some intuitive difficulties, but nonetheless the general idea of applying formal semantics defining the meaning of deontic notions on the basis of the way that complex actions are constructed from basic ones is important for further development of the field.

Finally, it was pointed out that deontic logic must be closely related to the theory of action. An interesting formulation of that idea is given in [35]. He concludes that there are branches of logic which are related to deontic logic to such extent that they may be regarded as extensions or offshoots of it. In particular, that applies to the formal theory of action and the logic of change.

The presentation of action logic introduced in the same paper of von Wright is also interesting and important for our further investigations. Actions are linked to and characterized by their results. Symbol $[p]x$ is used to express the fact that action x results in state p. Then, deontic notions are applied to actions via states, which are the results of the actions.

In such a presentation, action theory and deontic logic are put in one system which for that reason can be regarded as a hybrid one. Segerberg, as we describe in details in the next section, divides it strictly, leaving the deontic part in the system itself and shifting action theory to the semantics of the system.

1.3 Segerberg's deontic logic

In [26], Segerberg proposes to study the properties of the standard deontic operators using the mathematical theory of boolean algebras. The basic idea behind Segerberg's work is to interpret actions as elements of a boolean algebra and deontic operators as sets of elements in this algebra; intuitively, deontic operators denote the set of elements that make them true. These sets satisfy some well-known properties: they are closed for boolean conjunction and boolean inclusion; that is, they are ideals of the corresponding algebra. As explained in the introduction, fields of sets are boolean algebras, and then, there is a, more or less, straightforward way of getting an intuitive semantics based on sets: actions are interpreted as sets of *outcomes*, and then the permission and prohibition operators are interpreted as sets of outcomes that fulfill some requirements; these conditions imply that these sets describe ideals in the underlying boolean algebra of sets; and so both approaches to the semantics are equivalent. In the following we introduce the syntax and semantics of Segerberg's logic with some remarks that will be useful in the next sections, the interested reader can find the details in [26].

Vocabularies are made up of a denumerable set of action letters: $\{a,b,c,\dots\}$[2], we consider two action constants **0** and **1**. Actions may be combined with the use of action operators: negation represented by an overline, parallel execution ($\sqcup$) and free choice (*sqcap*). Atomic formulae are $\mathbf{Perm}(\alpha)$ (α *is allowed*), $\mathbf{Forb}(\alpha)$ (α *is forbidden*) and $\alpha = \beta$ (α *and* β *denote the same action*). We also have the standard propositional combinators: If φ and ψ are formulae, then $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ and $\neg\varphi$ are formulae. There are two equivalent ways of providing the semantics of this logic: one is interpreting actions as elements of a boolean algebra, the other one is by interpreting them as subsets of a set of possible outcomes. Let us introduce both semantics.

Consider structures of the form $\mathscr{A} = \langle A, \times, +, -, 0, 1, F, P\rangle$, where $\langle A, \times, +, -, 0, 1\rangle$ is a boolean algebra, F and P are ideals of this algebra and $F \cap P = \{0\}$ (i.e., they are disjoint ideals). We can define a valuation function, which maps actions to elements of the boolean algebra, as follows:

- $v(\mathbf{0}) = 0$.
- $v(\mathbf{1}) = 1$.
- $v(\alpha \sqcap \beta) = v(\alpha) \times v(\beta)$.
- $v(\alpha \sqcup \beta) = v(\alpha) + v(\beta)$.
- $v(\overline{\alpha}) = -v(\alpha)$.

Using v we define a satisfaction relationship $\vDash_A$ between boolean algebras, valuation functions, and formulae, as follows:

- $\mathscr{A}, v \vDash_A \alpha = \beta \iff v(\alpha) = v(\beta)$.
- $\mathscr{A}, v \vDash_A \mathbf{Forb}(\alpha) \iff v(\alpha) \in F$.
- $\mathscr{A}, v \vDash_A \mathbf{Perm}(\alpha) \iff v(\alpha) \in P$.

[2] In [26], these letters are called event letters, since this terminology may cause some confusion with the meaning given to the word *event* in other related logics, we call them action letters.

- $\mathscr{A}, v \vDash_A \varphi \wedge \psi \Longleftrightarrow \mathscr{A}, v \vDash_A \varphi$ and $\mathscr{A}, v \vDash_A \psi$.
- $\mathscr{A}, v \vDash_A \varphi \vee \psi \Longleftrightarrow \mathscr{A}, v \vDash_A \varphi$ or $\mathscr{A}, v \vDash_A \psi$ or both.
- $\mathscr{A}, v \vDash_A \neg\varphi \Longleftrightarrow$ not $\mathscr{A}, v \vDash_A \varphi$.
- $\mathscr{A}, v \vDash_A \varphi \rightarrow \psi \Longleftrightarrow$ not $\mathscr{A}, v \vDash_A \varphi$ or $\mathscr{A}, v \vDash_A \psi$, or both.

We say that $\vDash \varphi$ (φ is algebraically valid) iff $\mathscr{A}, v \vDash \varphi$ for every deontic action algebra $\mathscr{A}$ and every valuation v. Furthermore, given a set of formulae Γ, we say that $\Gamma \vDash \varphi$, if for every valuation v and every algebra $\mathscr{A}$, we have that, if $\mathscr{A}, v \vDash \psi$, for every $\psi \in \Gamma$, then $\mathscr{A}, v \vDash_A \varphi$.

Another interpretation of deontic operators is obtained by using set theory, we say that a structure $\mathscr{F} = \langle U, Ill, Leg \rangle$ is a deontic action frame (or deontic model) if U is a set and $Ill, Leg \subseteq U$ are two subsets of U such that $Ill \cap Leg = \emptyset$. We can think of U as the set of all possible outcomes. In this setting, the set Leg is the set of legal outcomes, and the set Ill is the set of illegal outcomes. A valuation is a function v from actions letters to the powerset of U. We can extend the definition of v using the usual set operators.

- $v(\mathbf{0}) = \emptyset$.
- $v(\mathbf{1}) = U$.
- $v(\alpha \sqcap \beta) = v(\alpha) \cap v(\beta)$.
- $v(\alpha \sqcup \beta) = v(\alpha) \cup v(\beta)$.
- $v(\overline{\alpha}) = U - v(\alpha)$.

We can define a relationship $\vDash$ between deontic models and formulae in a similar way that we defined $\vDash_{\mathscr{A}}$; we only introduce definitions for the deontic operators, the other ones are as usual.

- $\mathscr{F}, v \vDash \mathbf{Perm}(\alpha) \Longleftrightarrow v(\alpha) \subseteq Leg$.
- $\mathscr{F}, v \vDash \mathbf{Forb}(\alpha) \Longleftrightarrow v(\alpha) \subseteq Ill$.

We say that $\vDash \varphi$ if $\mathscr{F}, v \vDash \varphi$ for every valuation v and model $\mathscr{F}$. Similarly, we define the relationship $\Gamma \vDash \varphi$ between formulae.

Segerberg proved that the two notions of validity coincide. We do not present the proof here, the interested reader can consult [26].

Theorem 1. *For every set of formulae Γ and formula φ, we have:*

$$\Gamma \vDash \varphi \Leftrightarrow \Gamma \vDash_A \varphi$$

The logic has a simple axiomatic system:

1. Axioms of boolean algebra for $=$.
2. Axioms for equality.
3. $\mathbf{Forb}(\alpha \sqcup \beta) \leftrightarrow \mathbf{Forb}(\alpha) \wedge \mathbf{Forb}(\beta)$.
4. $\mathbf{Perm}(\alpha \sqcup \beta) \leftrightarrow \mathbf{Perm}(\alpha) \wedge \mathbf{Perm}(\beta)$.
5. $\alpha = 0 \leftrightarrow (\mathbf{Forb}(\alpha) \wedge \mathbf{Perm}(\alpha))$.

The unique deduction rule is the ancient *modus ponens*. If we have a proof (in the standard sense) of a formula φ, we say that $\vdash \varphi$; we also use this notation when we assume φ as an extra axiom. Note that axioms 3 and 4 state that prohibition and permission form ideals, while the last formula says that they denote disjoint sets. Using Lindenbaum-Tarski algebras we can prove the (strong) completeness of this system:

Theorem 2. $\Gamma \vdash \varphi \Leftrightarrow \Gamma \vDash \varphi$.

We do not reproduce the proof of this theorem, but it can be found in [26]. Let us explain the main technique used for the proof, since it will be useful in the next sections. Given a maximal consistent set of formulae Σ, we can define a relation of equivalence between actions, as follows:

$$\alpha \equiv_\Sigma \beta \Longleftrightarrow (\alpha = \beta) \in \Sigma$$

Since Σ is maximal, it is straightforward to prove that it is closed for the axiomatic system presented above, and therefore $=$ is an equivalence relation. Each action has an associated equivalence class:

$$\alpha_\Sigma = \{\beta \mid \alpha = \beta \in \Sigma\}$$

Using these ideas we can define the following algebra (the so-called Lindenbaum-Tarski algebra):

$$\langle \Delta/\Sigma, \sqcap_\Sigma, \sqcup_\Sigma, -_\Sigma, 0_\Sigma, 1_\Sigma, P_\Sigma, F_\Sigma \rangle$$

where:

- $\Delta/\Sigma = \{\alpha_\Sigma \mid \alpha \text{ is an action }\}$, is the set of equivalence classes of actions.
- $\alpha_\Sigma \sqcap_\Sigma \beta_\Sigma = (\alpha \sqcap \beta)_\Sigma$.
- $\alpha_\Sigma \sqcup_\Sigma \beta_\Sigma = (\alpha \sqcup \beta)_\Sigma$.
- $-_\Sigma \alpha_\Sigma = (-\alpha)_\Sigma$.
- $P_\Sigma = \{\alpha_\Sigma \mid \mathbf{Perm}(\alpha) \in \Sigma\}$.
- $F_\Sigma = \{\alpha_\Sigma \mid \mathbf{Forb}(\alpha) \in \Sigma\}$.

This algebra is a model for the set Σ, and therefore this proves the strong completeness of the system w.r.t. the algebraic models; to prove the completeness w.r.t. deontic models it is necessary to use the stone representation theorem to obtain a canonical model. Notice that the deontic operators induce ideals on the Lindebaum-Tarski algebra; these ideals are then used for defining the model. The Lindenbaum-Tarski construction will be useful for proving the completeness of related logics in section 1.4.

An important principle in jurisprudence (and therefore in deontic logic) is the so-called *Closure Principle*: *what is not forbidden is allowed*. Note that this principle is not a theorem of the system shown above. Because of this, Segerberg calls this logic *Basic Open Deontic Logic* (or BOD for short). The non-validity of the closure principle in this logic can be proven by inspecting the deontic models where we may have some outcomes that do not belong to *Ill* or *Leg*. Deontic logics that satisfy the closure principle are called *closed*, one is tempted to add the following restriction to models to obtain a closed logic: $U = Ill \cup Leg$, which seems to guarantee the closure principle; however, as shown in [27], these kinds of models are equivalent to the standard models (that is, they satisfy the same formulae in BOD). This seems surprising at first sight; however, this is a consequence of the impossibility of capturing individual outcomes using terms – action terms denote sets of outcomes, and the syntactical construction of the logic do not allow us to distinguish between singleton sets and sets with many elements. In section 1.4, we review some logics where it is possible to assert that individual outcomes are either permitted or forbidden. A possible solution to this issue is proposed by Segerberg using the following axiomatic schema:

$$\mathbf{Forb}(a) \vee \mathbf{Perm}(a) \qquad \text{(being } a \text{ an action letter)} \tag{1.1}$$

or, equivalently:

$$\neg\mathbf{Forb}(a) \rightarrow \mathbf{Perm}(a) \tag{1.2}$$

However, as stated in [31], this axiom induces some problems. Let us, for example, consider two actions *smoke* and *driving*. We may say that:

$$\vdash smoke \sqcap driving \neq \emptyset$$

That is, driving while smoking is possible. Suppose now that driving is allowed, this fact is formalized as follows: $\vdash \mathbf{Perm}(driving)$. But, since $\vdash driving \sqcap smoke \sqsubseteq driving$, using the axioms we get:

$$\vdash \mathbf{Perm}(driving \sqcap smoke)$$

by formula 1.1 and the fact that $smoke \neq \emptyset$ we get:

$$\vdash \mathbf{Perm}(smoke)$$

Summarizing, we get the following property:

$$\alpha \sqcap \beta \neq \emptyset \wedge \mathbf{Perm}(\alpha) \rightarrow \mathbf{Perm}(\beta) \tag{1.3}$$

which is not intuitively true. In section 1.4 we introduce some related logics that intend to tackle this issue.

It is possible to define other operators using permission and prohibition. One operator that is important in deontic logic is *obligation*; there are at least two ways of defining obligation in Segerberg's logic:

- $\mathbf{Obl}_P(\alpha) = \neg\mathbf{Perm}(\overline{\alpha})$.
- $\mathbf{Obl}_F(\alpha) = \mathbf{Forb}(\overline{\alpha})$.

The first one uses permission to define obligation, and the second one uses the prohibition operator. Intuitively, the first version of obligation says that an action is obligatory if and only if doing any other action is not allowed. In contrast, the second one says that an action is obligatory when it is forbidden to perform an alternative action. Let us write the satisfaction condition for the two versions of obligation:

- $\mathscr{F}, v \vDash \mathbf{Obl}_P(\alpha) \Longleftrightarrow U - v(\alpha) \nsubseteq Leg$.

- $\mathscr{F}, v \vDash \mathbf{Obl}_F(\alpha) \Longleftrightarrow U - v(\alpha) \subseteq Ill$.

A problematic issue with the first version of obligation (as already noted in [31]) is that strict refinements of forbidden actions are forbidden and obligatory at the same time, that is:

$$\alpha \sqsubseteq \beta \wedge \mathbf{Forb}(\beta) \wedge \alpha \neq \beta \rightarrow \mathbf{Forb}(\alpha) \wedge \mathbf{Obl}_P(\alpha)$$

For example, suppose the following statements:

- $\vdash \mathbf{Forb}(kill)$ (*it is forbidden to kill*).
- $\vdash kgently \sqsubseteq kill$ (*killing gently is a way of killing*).
- $\vdash kgently \neq kill$ (*there are some ways of killing that are not gentle*).

From these statements we can deduce: $\vdash \mathbf{Forb}(kgently) \wedge \mathbf{Obl}_P(kgently)$, the first part of the formula is intuitively true, but the second one does not fit with the intuitions: from the prohibition to kill we obtain that we are obliged to kill gently. This is a variation of the well-known paradox of the gentle killer, though no contrary-to-duty reasoning is involved in this case.

Let us take a look at the second version of obligation. Note that this version of obligation makes true the so-called Ross' paradox:

$$\mathbf{Obl}_F(\alpha) \rightarrow \mathbf{Obl}_F(\alpha \sqcup \beta)$$

which can be interpreted by saying: if you are obliged to send a letter, then you are obliged to send a letter or to burn it; which contradicts the common sense. Summarizing, the two versions of obligations described above do not capture the intuitive properties surrounding the concept of duty. In the next section we investigate other ways of defining obligation to avoid the problems explained above.

Segerberg presents his deontic logic of action just in a short paper. However, from today's perspective its content is important as well as inspiring. To sum up Segerberg's contribution to deontic action logic and his position towards problems occurring in it, let us point out the following issues.

- Segerberg's system is based on an action theory more sophisticated than truth value tables (as in Kalinowski's works); as a result, a deontic qualification of complex actions is not a simple function of generators. Thus, deontic qualification is essentially connected with complex actions.
- Segerberg introduces a novel semantics (defined using a domain of outcomes). He stresses the inspiration received from von Wright's paper [35], but in his paper he performs a strict separation between the axiomatic system and the semantics.
- Permission and forbiddance are not inter-definable in Segerberg's system. That creates the opportunity to discuss problems of openness and closeness of deontic action logic.
- Segerberg uses an infinite algebra of actions. Later works show that finite structures seems to be sufficient and much more handy.
- There is no operator corresponding to sequence of actions. Many things become much more interesting, but also complicated, when this combinator is introduced.

We point out some ways of introducing it in the deontic context presented in later works.

- In Segerberg's paper obligation is a defined notion. However, both definitions given in it leads to some counterintuitive consequences. We shall discuss the issue of obligation in more details in the next section.

1.4 Contemporary deontic action logics and boolean algebra

Several deontic logics with boolean operators have been proposed since the work of Segerberg. We distinguish between two kinds of logics; first, those logics that interpret deontic operators as sets of events/outcomes that fulfill these operators, among these logics we can cite those of Castro and Maibaum [7], R. van der Meyden [22] and Fiadeiro and Maibaum [9] as well as the work of Trypuz and Kulicki [31] enriching Segerberg's logic to obtain a more appealing version of obligation. On the other hand, the other kinds of logics are related to Dynamic Logic [13], this approach was initiated by J. J. Meyer in [23]; in this seminal work, Meyer relates modalities with deontic operators using violation markers. This line of research was followed by J. Broersen in his thesis [6], and by other authors. These works are related with Boolean Modal Logic defined by Gargov and Passy in [11], many of the properties of Dynamic Deontic Logics are inherited from the corresponding properties of Boolean Modal Logic, we present the details below. All these logics have a common feature of having terms for actions as well as operators to combine them; deontic operators can be used to state prescriptions over these action terms.

1.4.1 Deontic Dynamic Logics

Dynamic logic was introduced by Harel in [13]. This logic makes use of the box and diamond modalities to express the concepts of necessity and possibility, respectively. The novel part is that we have an infinite number of action letters; actions are combined with modalities to express the notion of causality, for example:

$$[a]\varphi$$

means: after executing action a, φ becomes true; on the other hand:

$$\langle a\rangle\varphi$$

says that it is possible to execute action a and finishing in a state of affairs where φ is true. Furthermore, we can combine actions as follows: if α and β are actions, then $\alpha;\beta$ is an action, α^* is an action and $\alpha \sqcup \beta$ is an action. Roughly speaking, ; expresses sequential composition (b is executed after a), $*$ expresses the Kleene operator: a is executed n times; and $\sqcup$ is the non-deterministic choice between actions α and β. The semantics of dynamic logic is given by models made up of a non-empty set of worlds W, a relationship $R_a \subseteq W \times W$ for each action letter a, and an interpretation function mapping propositional letters to sets of worlds. In this setting, the action combinators are interpreted as usual relational operators. For example, the sequential composition is interpreted as the relational composition; the non-deterministic choice is interpreted as the relational union and the star operator is interpreted as the reflexive-transitive closure of relations. There exist sound and complete axiomatic systems for dynamic logic; however, as a consequence of the fact that the star operator is not elementary, the logic is not compact – the details can be found in [13].

One important variation of dynamic logic is the so-called Boolean Modal Logic [11] (or BML), where the boolean operators are used for combining actions; the semantics of these operators is given by means of the usual relational constructions. One important point about this logic is that the complement enables the introduction of the *window operator*, an operator that allows us to inspect any state related or not to the actual state, some authors have pointed out that this operator violates in some sense the principle of locality implicit in modal logics, see [4]. BML has sound and complete axiomatic systems, though this logic is not strongly complete nor compact.

John Jules Meyer uses the constructions of Dynamic Logic to define what he calls *Dynamic Deontic Logic* [23]; In this work, deontic constructions are reduced to dynamic logic constructions using a violation constant which indicates that a violation has been produced. Meyer proposes to use the following combinators: ; (composition), $\sqcup$ (non-deterministic choice), $\sqcap$ (parallel execution), and $-$ (alternative action). An algebra of actions, resembling boolean algebras, is proposed for these action combinators; however, the properties of this algebra of actions are not investigated by the author (indeed it is possible to prove that there is not decidable axiomatizations for these kinds of algebras [21]). Using modalities, Meyer defines:

$$\mathbf{Forb}(\alpha) \leftrightarrow [\alpha]\mathbf{v}.$$

That is, an action is forbidden if and only if every execution of this action yields a violation. Using prohibition, Meyer defines the rest of the deontic predicates:

- $\mathbf{Obl}(\alpha) \leftrightarrow \mathbf{Forb}(\overline{\alpha})$ (obligation) and,
- $\mathbf{Perm}(\alpha) \leftrightarrow \neg\mathbf{Forb}(\alpha)$ (permission).

Broersen [6] called this approach *goal oriented norms* since, for evaluating the truth value of a deontic predicate, only the resulting state of an action is important and not what happens during its execution. In [6] the boolean operators are used in combination with the deontic operators and the modalities; in this setting, Broersen obtains a sound and complete dynamic deontic logic with boolean operators; however, this logic is not compact.

Several criticisms have arisen to this approach. For example in [22], the following formula is exhibited as a paradox of dynamic deontic logic: $\langle\alpha\rangle\mathbf{Perm}(\beta) \rightarrow \mathbf{Perm}(\alpha;\beta)$, which can be read as *if after shooting the president it is allowed to remain silent, then it is allowed to shoot the president and remain silent,* which is undoubtedly undesirable; these kinds of problems are inherent in goal oriented norms, Broersen has proposed the so-called process-oriented norms to deal with this problem, see [6] for the details.

In [1] these ideas are used to establish a more serious paradox: $\mathbf{Forb}(\alpha) \rightarrow [\alpha]\mathbf{Obl}(\beta \sqcap \overline{\beta})$, i.e., after executing a forbidden action, we are obliged to perform an impossible action, which is not intuitively true. In spite of these facts, Meyer's approach is interesting since in deontic dynamic logic a clear division between predicates and actions is established and, as Meyer argues, some paradoxes vanish in this approach, mainly since here we have a notion of time or state change. Moreover, some problematic statements, like nested deontic constraints, are no longer expressible. In the following section we introduce another branch of deontic action logic, initiated from the ideas of Segerberg, in which deontic operators are not captured by using modalities, instead an algebra is used to formalize the concept of norm.

1.4.2 Deontic Logics based on Atomic Boolean Algebras

Segerberg used boolean algebra to give the semantics of deontic operators; in [7, 31] a variation of this approach is taken: the set of action letters is considered finite and therefore the underlying algebra of actions becomes atomic. Atomic boolean algebras have some good properties, from the topological point of view, the atoms allow us to refer to the points of the underlying space: there is a one-to-one mapping between the set of atoms of a boolean algebra and the set of its maximal ideals (or ultrafilters); the maximal ideals (or ultrafilters) can be thought of as points of the field of sets which is isomorphic to the boolean algebra (by the Stone theorem). Roughly speaking, we can refer in the language to the most specific actions that can be executed. For example, consider that we have two possible actions: *driving* and *drinking*, if we abstract ourselves from the other possible actions, we obtain the (canonical) boolean algebra of figure 1.1. Note that the atoms in this algebra are:

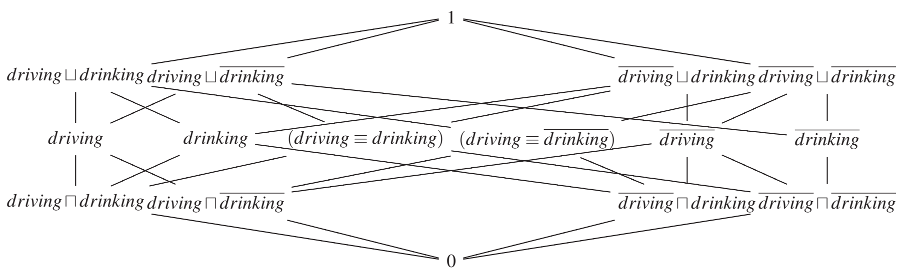


Fig. 1.1 Canonical Boolean Algebra for three actions

$driving \sqcup drinking$, $driving \sqcup \overline{drinking}$, $\overline{driving} \sqcup drinking$, $\overline{driving} \sqcup \overline{drinking}$. Every atom can be identified with an ultrafilter. For example, the atom $driving \sqcap drinking$ can be identified with the filter shown in figure 1.2. This filter can be thought of as

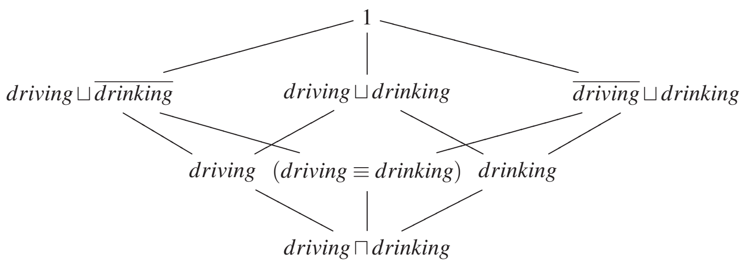


Fig. 1.2 Filter identified with atom $driving \sqcap drinking$

stating a set of weakly allowed actions. In the same way, coatoms identify maximal ideals, and therefore sets of strongly allowed actions. Consider, for example, the coatom: $drinking \sqcup driving$, in this case we obtain the ideal shown in figure 1.3. This ideal may, for example, identify a set of strongly permitted actions. Let us

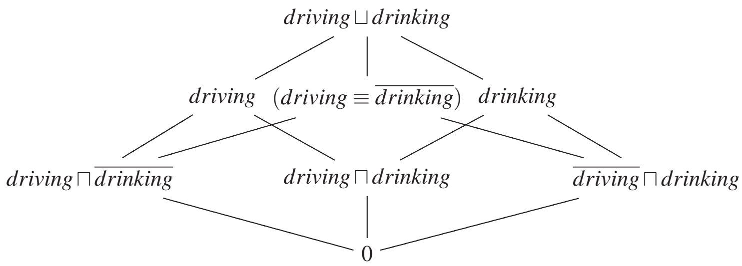


Fig. 1.3 Ideal corresponding to coatom $drinking \sqcap driving$

note that atoms are monomials made up of atomic letters (or negation of them) composed by the $\sqcap$ operator; that is, it is straightforward to determine which action terms denote atoms in the corresponding boolean algebra and which do not. Let us note that, if we add the restriction $driving \sqcup drinking = 1$, then the diagrams above can be simplified, for example, the action $\overline{driving} \sqcap \overline{drinking}$ is an impossible action (that is, it is equal to 0). In some sense, this restriction says that no other actions are possible. This view of restricting ourselves to a finite number of actions has many interesting consequences, and, of course, triggers philosophical questions. One may think that the number of possible actions is potentially infinite; however, usually we are interested in reasoning about a particular set of actions, and a finite set (which may be very large) seems to be enough in most of the scenarios. No much expressivity is lost when the set of actions is restricted to a finite set, but the possibility of talking about atoms is gained, and this allows us to express interesting properties about the logic.

We shall first discuss some remarks about the semantics in the finite case. The semantics is given by means of structures: $\langle Out, Ill, Leg \rangle$, similar to the ones used by Segerberg. Note that atomic action terms are intended to express actions where no ambiguity is left, that is, each atomic action describes the actions letters involved during the execution of the action; an intuitive semantic restriction (in this case) is that atomic action terms denote at most one outcome; roughly speaking, these actions are deterministic. This restriction can be added as follows:

$$|\mathscr{I}(\delta)| \leq 1 \tag{1.4}$$

where $|-|$ denotes the cardinality of sets, and δ denotes an action term that is an atom in the boolean algebra of actions. The basic axioms of this logic are the following:

- $\mathbf{Perm}(\alpha \sqcup \beta) \equiv \mathbf{Perm}(\alpha) \wedge \mathbf{Perm}(\beta)$.
- $\mathbf{Forb}(\alpha \sqcup \beta) \equiv \mathbf{Forb}(\alpha) \wedge \mathbf{Forb}(\beta)$.
- $\alpha = 0 \equiv \mathbf{Forb}(\alpha) \wedge \mathbf{Perm}(\alpha)$.

Of course, we have the usual axioms for equality and boolean algebras. This system is equivalent to Segerberg's system (BOD). In addition to the standard operators we can define the weak version of them:

- $\mathbf{Perm}_w(\alpha) = \neg\mathbf{Forb}(\alpha)$
- $\mathbf{Forb}_w(\alpha) = \neg\mathbf{Perm}(\alpha)$

Below we investigate the interpretation of the weak deontic operators.

We may use the atoms to introduce some further axioms. In the following we analyze the possible extensions of BOD, we follow the ideas of [31] to classify the systems.

1.4.3 Extensions of BOD

1.4.3.1 The Basic Closed System

As remarked above, Segerberg points out that closeness in BOD can be introduced by the following axiom:

$$\mathbf{Forb}(a_i) \vee \mathbf{Perm}(a_i) \qquad \text{for every action letter } a_i \tag{1.5}$$

as we shown in section 1.3, this axiom has some paradoxical consequences, implying that actions that can be performed together must have the same deontological status. Furthermore, when we have a finite number of actions: $a_0, \ldots, a_n$, the atomic term:

$$\overline{a_0} \sqcap \ldots \sqcap \overline{a_n} \tag{1.6}$$

deserves special attention; note that this term can be interpreted as saying that no action of the actual agent is executed; moreover, this action may be thought of as denoting some behavior of an external agent. Let us note that formula 1.5 is not expressible enough to state that action term 1.6 is allowed or forbidden. Note that, if we have an infinite number of actions, there is no way to capture the notion of external actions, though this might be achieved by dividing the actions into internal and external ones, but this complicates the syntax of the logic. If we want to ensure closedness in the finite case, we must add the following axiom:

$$\mathbf{Perm}(\overline{a_0} \sqcap \ldots \sqcap \overline{a_n}) \vee \mathbf{Forb}(\overline{a_0} \sqcap \ldots \sqcap \overline{a_n}) \qquad \text{(being } a_0, \ldots, a_n \text{ all the action letters.)} \tag{1.7}$$

We call the system BOD+Axiom 1.5 **Basic Closed System** (BCS). In this system, any atomic action term δ is allowed or forbidden; that is, we have the following theorem: $\vdash \mathbf{Perm}(\delta) \vee \mathbf{Forb}(\delta)$

1.4.3.2 The Atomic Closed System

It is possible to use the atoms to state the closeness of the system at a low level, that is, we can state that the atomic actions are allowed or forbidden:

$$\mathbf{Forb}(\delta) \vee \mathbf{Perm}(\delta) \qquad \text{for every atomic term } \delta \tag{1.8}$$

This axiom, in contrast to axiom 1.5, avoids the paradox expressed by formula 1.3; note that, if two atomic actions have a non-empty intersection, then they are the same action. This axiom is adequate to models satisfying the following principle:

$$\mathscr{E} = Ill \cup Leg$$

We call the system BOD+Axiom 1.8 **Atomic Closed System** (ACS). Note that in this system the term $\overline{a_0} \sqcap \ldots \sqcap \overline{a_n}$ may denote some outcomes that can be interpreted as outcomes of external actions.

1.4.3.3 The Standard Atomic Closed System

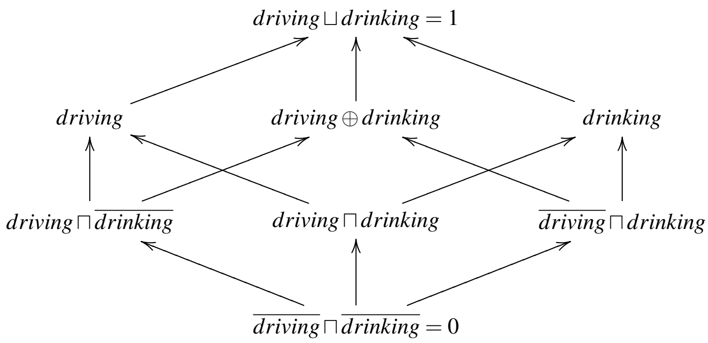
As we remarked above, the action $\overline{a_0} \sqcap \ldots \sqcap \overline{a_n}$ may be thought of as the action of doing nothing; however, if we consider a special action *skip* to denote this particular event, then the action $\overline{a_0} \sqcap \ldots \sqcap \overline{a_n}$ denotes an impossible action; that is, we have:

$$\overline{a_0} \sqcap \ldots \sqcap \overline{a_n} = 0 \tag{1.9}$$

or by duality:

$$a_0 \sqcup \ldots \sqcup a_n = 1 \tag{1.10}$$

We call the system ACS+Axiom 1.9 **Standard Atomic Closed System** or SACS. This system is presented in [7] under the name DPL, and in [31] is called DAL[5]. There are some interesting remarks about this logic; first, let us note that the Hasse diagram of the canonical boolean algebra for two actions (*driving* and *drinking*).



Where $driving \oplus drinking = (driving \sqcap \overline{drinking}) \sqcup (\overline{driving} \sqcap drinking)$ is the exclusive or between *drinking* and *driving*. Note that, the definition of $\mathbf{Perm}_w(-)$ together with axiom 1.10, implies that the weak permission is semantically interpreted as the union of filters defined by the atoms which are strongly permitted. Weak permission does not define a filter since it is not closed for $\sqcap$.

1.4.3.4 The Relationship between BOD, BCS, ACS, SACS, SCS

The relationship between these logics is shown by the diagram in figure 1.4 [31], where an arrow from one system to another means that all the theorems of the source

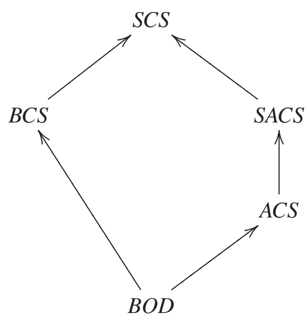


Fig. 1.4 Relation between the different logical systems

system are theorems in the target system. The picture can be completed if we add subsystems along the diagram; let us remark that the system *SCS* seems to be too strong to be accepted, as it is shown by formula 1.3, the other systems can be accepted or not, depending on the level of closure that we intend to capture.

1.4.4 The Obligation Operator

The formalization of the obligation has been controversial from the beginnings of deontic logic; in particular, in deontic action logics there exist several variations of the concept of obligation, in this section we review the usual ones. Meyer defines obligation as follows:

$$\mathbf{Obl}_F(\alpha) = \mathbf{Forb}(\overline{\alpha}) \tag{1.11}$$

That is, an action is obligatory iff doing any alternative action is forbidden. Obligation is defined as the complement of an ideal (prohibition) and therefore the interpretation of this operator defines a filter in the underlying boolean algebra. As a consequence, this version of obligation has the following properties:

- $\mathbf{Obl}_F(1)$
- $\mathbf{Obl}_F(\alpha \sqcap \beta) \equiv \mathbf{Obl}_F(\alpha) \wedge \mathbf{Obl}_F(\beta)$

Moreover, this obligation holds the so-called Ross' paradox:

$$\mathbf{Obl}_F(\alpha) \to \mathbf{Obl}_F(\alpha \sqcup \beta)$$

which admits the following reading: *if you are obliged to send a letter, then you are obliged to send a letter or to burn it.* Note that an obliged action (following this definition) may have some illegal outcomes, that is, an obliged action may not be allowed; this does not satisfy the principle: $\mathbf{Obl}(\alpha) \to \mathbf{Perm}(\alpha)$, which may be desirable in some contexts.

Another definition of obligation can be obtained by using the permission, as follows:

$$\mathbf{Obl}_P(\alpha) = \neg\mathbf{Perm}(\overline{\alpha})$$

Roughly speaking, an action is obligatory (following this definition) when some outcomes of alternative actions are not allowed. This operator has the following properties:

- $\mathbf{Obl}_P(\alpha \sqcup \beta) \equiv \mathbf{Obl}_P(\alpha) \vee \mathbf{Obl}_P(\beta)$.
- $\mathbf{Obl}_P(\alpha \sqcup \beta) \to \mathbf{Obl}_P(\alpha) \wedge \mathbf{Obl}_P(\beta)$.

As remarked in [31], a problematic property of this variation of obligation is the following one:

$$\mathbf{Forb}(\beta) \wedge \alpha \sqsubseteq \beta \wedge \alpha \neq \beta \to \mathbf{Obl}_P(\alpha)$$

That is, specific ways of performing forbidden actions are obligatory, which is paradoxical.

Let us present another possible definition of obligation, introduced in [7]. The definition is as follows:

$$\mathbf{Obl}_F^P(\alpha) = \mathbf{Perm}(\alpha) \wedge \mathbf{Forb}(\overline{\alpha})$$

Roughly speaking, an action is obligeatory if it is allowed and any alternative action is forbidden. This definition does not hold the Ross' paradox, moreover it satisfies some intuitive properties [7]:

- $\mathbf{Obl}_F^P(\alpha) \rightarrow \mathbf{Perm}(\alpha)$
- $\mathbf{Obl}_F^P(\alpha) \wedge \mathbf{Obl}_F^P(\overline{\alpha}) \equiv (\alpha = 0)$

However, this definition of obligation satisfies the following property (called *extensionality* in [31]):

$$\mathbf{Obl}_F^P(\alpha) \wedge \mathbf{Obl}_F^P(\beta) \rightarrow \alpha = \beta$$

That is, only one action can be obligatory per time; this seems paradoxical as we can devise scenarios where this is not the case.

Trypuz and Kulicki have proposed another version of obligation which intends to improve the definitions of obligation given above. The idea is to add a new set *Req* of *required* outcomes, and therefore we can introduce the obligation as a new operator as follows:

$$\mathbf{Obl}_N(\alpha) \Longleftrightarrow \mathscr{I}(\alpha) \subseteq Req$$

We may add the requirement that *Req* is not empty: $Req \neq \emptyset$. The properties of this new version of obligation are the following:

- $\mathbf{Obl}_N(\alpha) \wedge \mathbf{Obl}_N(\beta) \rightarrow \mathbf{Obl}(\alpha \sqcap \beta)$
- $\neg\mathbf{Obl}_N(0)$

Of course, if we want to obtain: $\mathbf{Obl}_N(\alpha) \rightarrow \mathbf{Perm}(\alpha)$, we should add the following requirement:

$$Req \subseteq Leg$$

However, the following principle cannot be proven for this version of obligation:

$$\mathbf{Obl}_N(\alpha) \rightarrow \mathbf{Forb}(\overline{\alpha})$$

To summarize, when we introduce the notion of atom in the basic logic we obtain several extensions of this logic, these extensions are obtained by adding different levels of closeness as well as different versions of obligation; it is not our intention to favor one deontic system over others, we leave this to the reader. In the following section we discuss some possible lines of future work; in particular, it seems interesting to extend deontic logics with boolean algebras with operators that also support the concept of atom and coatom.

1.4.5 A Deontic Logic Built on Synchronous Kleene Algebra

The language of DAL built on synchronous Kleene algebra

In a recent paper [25], another system (on the technical side inspired by the use of algebraic structures in the papers of Segerberg [26] and Castro and Maibaum [7]) based on intuitions similar to the system of Meyer from [23] is presented. Formally, the space of actions is represented there by an algebraic structure called synchronous Kleene algebra, defined in [24]. Such algebra differs from boolean algebra by having the operator of sequential composition on its elements instead of negation (complement). A kind of action complement is introduced into the system by definition, as a non-primitive notion. The work opens new possibilities for deontic action logic offering a new, interesting semantic tool.

Moreover, contratry-to-duty obligations, that are not expressible in the earlier mentioned systems, are introduced in the form of a *reparation* connected with obligation and prohibition. Thus, formulas $\mathbf{Obl}_{\mathscr{C}}(\alpha)$ and $\mathbf{Forb}_{\mathscr{C}}(\alpha)$ state respectively that α is obligatory (forbidden) and if an agent breaks such a norm it is bound by another norm expressed by $\mathscr{C}$, which is a reparation. Formulas $\mathbf{Obl}_{\perp}(\alpha)$ and $\mathbf{Forb}_{\perp}(\alpha)$ are understood as an absolute obligation and forbiddance.

Formally, we can define the language of the system as follows[3]:

$$\alpha := a \mid 0 \mid 1 \mid \alpha \sqcap \alpha \mid \alpha \sqcup \alpha \mid \alpha ; \alpha$$

$$\mathscr{C} := \perp \mid \mathbf{Perm}(\alpha) \mid \mathbf{Forb}_{\mathscr{C}}(\alpha) \mid \mathbf{Obl}_{\mathscr{C}}(\alpha) \mid \mathscr{C} \rightarrow \mathscr{C}$$

where a is an element of a finite set A of basic actions.

Let further $A^{\sqcap}$ be the set of actions composed from basic actions from A using only $\sqcap$ operator. Intuitively, the set $A^{\sqcap}$ contains actions that are parallel executions of an arbitrary number of basic actions. By analogy to boolean algebra of actions, we will call the elements of $A^{\sqcap}$ *quasiatoms*[4]. The difference is that atoms of BA can be described by parallel executions of basic actions or negation (complement) of them. Kleene algebra lacks boolean negation and quasiatoms contain only 'positive' parts of atoms. At this point we do not prejudge the semantic relation between atoms of BA and quasiatoms, this can be figured out from the formal semantics of the system. We shall write that quasiatom α is contained in quasiatom β ($\alpha \subseteq \beta$), when the set of basic atoms from which α is composed is contained in the set of basic actions from which β is composed.

0 is interpreted, as in boolean algebra of actions, as an impossible action. In contrast 1 is understood differently, as 'skip' or 'doing nothing'.

---

[3] We omit propositional constants originally used in [25].

[4] In [25] such formulas are called $\times$-formulas.

Axioms of synchronous Kleene algebra

The following axioms of boolean algebra listed in our Introduction (applied to the language of the system) are also axioms of synchronous Kleene algebra:

- Absorption of Zero,
- Identity Laws,
- Commutativity Laws,
- Associativity Laws,
- Distributivity of $\sqcup$ over $\sqcap$,
- Idempotency of $\sqcup$.

Absorption of 1 does not hold since, as mentioned above, 1 has a different meaning here than in boolean algebra. The system does not include idempotency of $\sqcap$. Instead of the latter law, the following weak idempotency of $\sqcap$ (idempotency for basic actions) is used:

If $a \in A$**, then** $\alpha \sqcap \alpha = \alpha$

The following formulas complete the axiomatization of equality in synchronous Kleene algebra of actions:

- $\alpha;(\beta;\gamma) = (\alpha;\beta);\gamma$ *(Associativity of* ;*).*
- $\alpha;1 = 1;\alpha = \alpha$ *(Identities of 1 with respect to* ;*).*
- $\alpha;0 = 0;\alpha = 0$ *(Absorption of zero with respect to* ;*).*
- $\alpha;(\beta \sqcup \gamma) = (\alpha;\beta) \sqcup (\alpha;\gamma)$ **and** $(\alpha \sqcup \beta);\gamma = (\alpha;\gamma) \sqcup (\beta;\gamma)$ *(Distributivity of* ; *over* $\sqcup$*).*
- **If** $\alpha, \beta \in A^{\sqcap}$**, then** $(\alpha;\gamma) \sqcap (\beta;\delta) = (\alpha \sqcap \beta);(\gamma \sqcap \delta)$ *(Weak distributivity of* $\sqcap$ *over* ;*).*

The system of deontic logic from [25] is defined semantically (no axiomatization for deontic notions is given). The following notions and facts are used to define a valid deontic proposition. We use the content of the definitions from [25], slightly changing the way they are presented there[5].

Canonical form

The inductive definition of canonical forms is the following:

(i) 0 is in canonical form.
(ii) If for all $i \in I$:
(1) either (a) $\beta^i = \alpha_1^i;\alpha_2^i$, (where $\alpha_1^i \in A^{\sqcap}$ and $\alpha_2^i \notin \{0,1\}$ is in canonical form), or
(b) $\beta^i = \alpha_1^i$, where $\alpha_1^i \in A^{\sqcap} \cup \{1\}$
and

[5] As the present paper has a character of a review we refrain from criticizing particular intuitions behind the system and proposing alternative solutions.

(2) for all $i, j \in I$ if $i \neq j$, then $\alpha_1^i \neq \alpha_1^j$,
then $\alpha = \sqcup_{i \in I} \beta^i$ is in canonical form.

Each α_1^i plays the role of a unique possible first step of compound action α – the first step of action β^1. Action α_1^i cannot be equal to 0, since in that case α would also be equal to 0. In case (a) it must be a quasiatom. In case (b), action β^1 is a one step action (α_1^i is its first and its last step). In that case α_1^i is a quasiatom or equals 1. Thus quasiatoms and 1 are in canonical form (when, in case (b), I is a singleton).

Each α_2^i is the rest of action β^i. Action α_2^i cannot equal 0 (for the same reasons as α_1^i) or 1 (because of identity of 1 w.r.t. ;).

For any action α there exists α' in canonical form s.t. $\alpha = \alpha'$ ([24] Th. 2.8).

Action complement

Action complement is not a principal combinator but it is a function defined inductively as follows.

(i) Complement of 0 is 1, complement of 1 is 0, in symbols $\overline{0} = 1$, $\overline{1} = 0$.

(ii) Let $\alpha \notin \{0, 1\}$ be an action in canonical form, i.e. $\alpha = \sqcup_{i \in I} \beta^i$, where for all $i \in I$ $\beta^i = \alpha_1^i$ or $\beta^i = \alpha_1^i; \alpha_2^i$ as in the definition of canonical form.

Let further X_1 be the set of α_1^i s.t. $i \in I$ and $\beta^i = \alpha_1^i; \alpha_2^i$ (β^i is not a one step action), $\overline{X_1} = \{\gamma \in A^{\sqcap} \mid \neg \exists_{i \in I}\ \alpha_1^i \subseteq \gamma\}$. Moreover, let δ^j $(j \in J)$ be all quasiatoms s.t. $\exists_{\alpha \in X_1}\ \alpha \subseteq \delta^j$ and $I_j \subseteq I$ be indexing set s.t. $I_j = \{i \in I \mid \alpha_1^i \subseteq \delta^j\}$.

Complement $\overline{\alpha}$ of action α is defined by the following equation:

$$\overline{\alpha} = \sqcup \overline{X_1} \ \sqcup \ \sqcup_{j \in J} (\delta^j; \overline{\sqcup_{i \in I_j} \alpha_2^i})$$

Intuitively, a complement of a multiple step action is a free choice between different ways of not doing the first step of the action and doing the first step, and different ways of not doing the other steps. A complement of an action cannot have more steps then the original action. That makes the construction finite.

Proposition 2.8 from [25] states that the complement operation returns a deontic action which is in canonical form.

Rooted tree

Let A be a set of basic actions. A rooted tree with labelled edges is an acyclic conected graph $\langle \mathscr{N}, \mathscr{E}, A \rangle$ with a designated node r. $\mathscr{N}$ is a set of nodes, $r \in \mathscr{N}$ is a designed node called root node. $\mathscr{E}$ is the set of directed labelled edges between nodes (in symbolical notation $m \xrightarrow{\alpha} n$ stands for the edge from node m to node n

with label α), where labels are taken from the set $2^A \cup \{\Lambda\}$.

Intuitively, nodes represent states and edges – actions that can lead from one state to another by performing an action specified by a label. Empty label represents skip action 1, label Λ represents the impossible action 0 and all the other labels represent quasiatoms built from the elements of the label. For that reason, we use the same variables for labels as for actions. Multiple edges starting from one node represent the free choice operator.

A path in the rooted tree is understood in a way usual for graphs. A path which cannot be extended (there is no edge starting from its last node) is called *final*. The final nodes on each final path are called *leaf nodes*. When an edge e is an element of the set of edges $\mathscr{E}$ of a tree T we shall write in short that e is an element of T $(e \in T)$.

Theorem 2.10 from [25] states that for any action in canonical form there exists a rooted tree corresponding to that action. For arbitrary action α we shall use the symbol $T(\alpha)$ to refer to the tree corresponding to the action in canonical form equal to α.

Normative structure

Let A be a set of basic actions. A normative structure is a triple $K = (\mathscr{W}, R_A, \rho)$, in which:

- $\mathscr{W}$ is a set of worlds;
- R_A is a function returning a labelled patrial accessibility function $R_\alpha : \mathscr{W} \longrightarrow \mathscr{W}$ for each set of basic actions $\alpha \subseteq A$;
- ρ is a marking function which marks each world with markers from the set $\{\circ_a, \bullet_a \mid a \in A\}$ in such a way that no world can be marked by both $\circ_a$ and $\bullet_a$ for any $a \in A$.

A pointed normative structure $\langle K, i \rangle$ is a normative structure with designated world i $(i \in \mathscr{W})$. As for trees, we shall call an element $e = s \xrightarrow{\alpha} s'$ of a partial accessibility function R_α also an element of K (symbolically: $e \in K$).

K is deterministic as for each set of basic actions there is at most one world connected by the relation. The relation informs us what actions can be executed in each world. Markers on the successor world inform us which actions are obligtory ($\circ_a$) and which are forbidden ($\bullet_a$). Marking function ρ marks each world for each basic action $a \in A$ with $\circ_a$, $\bullet_a$ or nothing, that means that actions leading to that world can be obligatory, forbidden or neutral.

Relationship between normative structures and rooted trees

For a tree $T = (\mathscr{N}, \mathscr{E}, A)$ and normative structure $K = (\mathscr{W}, R_A, \rho)$ let $\mathscr{S} \subseteq \mathscr{N} \times \mathscr{W}$ be the *simulation* relation of the tree node by the world of the structure s.t.:
$t\mathscr{S}s$ iff the following two conditions hold:

(i) for every edge $t \xrightarrow{\alpha} t' \in T$ there exists an element of a labelled accesibility relation $s \xrightarrow{\alpha'} s' \in K$ s.t. $\alpha \subseteq \alpha'$ and $t'\mathscr{S}s'$;
(ii) for every edge $t \xrightarrow{\alpha'} t' \in T$ and every element of a labelled accesibility relation $s \xrightarrow{\alpha'} s' \in K$ if $\alpha \subseteq \alpha'$, then $t'\mathscr{S}s'$.

We shall write that a tree T with root r is simulated by a normative structure K w.r.t. a world s ($T\mathscr{S}_sK$) if and only if $r\mathscr{S}s$.

In the definition, the label of the edge α of the tree is included in the label α' of the accessibility relation in the normative structure. Prisacariu and Schneider motivate this by the idea that, respecting an obligatory quasiatomic action constructed from elements of α means executing any quasiatomic action in which it is included. Intuitively a tree representing an action is represented by a normative structure if every possible way of executing any step of the action allows to execute another step of the action. Because the inclusion of α in α' is used, any step can be executed in parallel with any other quasiatomic action.

This simulation relation can be strengthened to a *strong simulation* by changing the conditions $\alpha \subseteq \alpha'$ in (i) and (ii) into the equivalence $\alpha = \alpha'$. Then, since K is a deterministic condition, (ii) is redundant. We shall use symbol $\mathscr{S}'$ for strong simulation. In this case, only the exact execution (with no other actions executed in parallel) of quasiatomic steps is considered.

The notion of simulation can be also weakened by dropping existential condition (i) from the definition. Such relation will be called *partial simulation* and it will be symbolically represented by $\widetilde{\mathscr{S}}$. In this case some steps of the action defining the simulated tree may not be executable, but if a step is executable, then the tree starting from the end of the step is partially simulated.

Now we define fragments of deontic structures, generated by rooted trees, which we shall call *simulating structure*[6] and *non-simulating reminder*.

Let T be a rooted tree, $K = \langle \mathscr{W}, R_A, \rho \rangle$ a deontic structure and $i \in \mathscr{W}$ a world s.t. $T\mathscr{S}_iK$.

$K^{T,i}_{sim} = (\mathscr{W}', R'_A, \rho')$ is a simulating structure w.r.t. T and i when it is the least sub-structure of K respecting the following conditions:
(i) $i \in \mathscr{W}'$;

[6] In [25] it is called maximal simulating structure.

(ii) if $t \xrightarrow{\alpha} t' \in T$ and $s \xrightarrow{\alpha'} s' \in K$ and $t\mathscr{S}s$ and $\alpha \subseteq \alpha'$, then $s' \in \mathscr{W}'$ and $s \xrightarrow{\alpha'} s' \in R'_A$;
(iii) $\rho' = \rho|\mathscr{W}'$.

$K^{T,i}_{rem} = (\mathscr{W}'', R''_A, \rho'')$ is a non-simulating reminder of K w.r.t. T and i when it is the least sub-structure of K respecting the following conditions:

(i) if $s \in K^{T,i}_{max}$ and there exist α' and s' s.t. $s \xrightarrow{\alpha'} s' \in K^{T,i}_{max}$ and $s \xrightarrow{\alpha} s'' \notin K^{T,i}_{max}$, then $s, s'' \in \mathscr{W}''$ and $s \xrightarrow{\alpha} s'' \in R''_A$;
(ii) $\rho'' = \rho|\mathscr{W}''$.

Validity

Now we are ready to define valid deontic formulae. The satisfaction of a deontic formula $\mathscr{C}$ w.r.t. a pointed normative structure $\langle K, i\rangle$ ($K, i \models \mathscr{C}$) is defined inductively as follows.

- $K, i \not\models \bot$
- $K, i \models \mathscr{C}_1 \to \mathscr{C}_2$ iff whenever $K, i \models \mathscr{C}_1$, then $K, i \models \mathscr{C}_2$
- $K, i \models \mathbf{Obl}_{\mathscr{C}}(\alpha)$ iff the following conditions hold:
  1. $T(\alpha)\mathscr{S}_i K$;
  2. if $t \xrightarrow{\beta} t' \in T(\alpha)$ and $s \xrightarrow{\beta'} s' \in K$ and $t\mathscr{S}s$ and $\beta \subseteq \beta'$ and $a \in \beta$, then $\circ_a \in \rho(s')$;
  3. if $s \xrightarrow{\beta'} s' \in K^{T(\alpha),i}_{rem}$ and $a \in \beta'$, then $\circ_a \notin \rho(s')$;
  4. if t is a leaf of a final path of $T(\overline{\alpha})$ starting from its root and $t\mathscr{S}^s s$, then $K, s \models \mathscr{C}$.
- $K, i \models \mathbf{Forb}_{\mathscr{C}}\alpha$ iff the following conditions hold:
  1. $T(\alpha)\widetilde{\mathscr{S}_i}K$;
  2. if σ is a final path of $T(\alpha)$ s.t. $\sigma\mathscr{S}_i K$ and $t \xrightarrow{\beta} t' \in \sigma$ and $s \xrightarrow{\beta'} s' \in K$ and $t\mathscr{S}s$ and $\beta \subseteq \beta'$ and $a \in \beta'$, then $\bullet_a \in \rho(s')$;
  3. if σ is a final path of $T(\alpha)$ starting from its root s.t. $\sigma\mathscr{S}_i K$ and t is a leaf of σ and $t\mathscr{S}s$, then $K, s \models \mathscr{C}$.
- $K, i \models \mathbf{Perm}(\alpha)$ iff the following conditions hold:
  1. $T(\alpha)\mathscr{S}_i K$;
  2. if $t \xrightarrow{\beta} t' \in T(\alpha)$ and $s \xrightarrow{\beta'} s' \in K$ and $t\mathscr{S}s$ and $\beta \subseteq \beta'$ and $a \in \beta$, then $\bullet_a \notin \rho(s')$

We say that $\mathscr{C}$ is satisfied in normative structure K ($K \models \mathscr{C}$) iff it is satisfied in every world of K. A deontic formula $\mathscr{C}$ is valid ($\models \mathscr{C}$) if it is satisfied in any deontic structure.

Let us now examine briefly the intuitive meaning of the definition of satisfaction. The first condition for obligation states that an obligatory action must be executable. The second one states that at each step all alternative executions of the step (defined by the free choice operator) are indeed obligatory. The third one states that no other possible alternative transition from any world in the normative structure is obligatory. Finally, the fourth condition states that, at the end of any alternative path in the normative structure (violating the obligation defined in the considered proposition), a proposition defining a reparation holds.

In the first conditions for obligation only weak simulation is used. Thus, the impossible action is regarded as forbidden. The second condition states that, if the considered action can be executed in a certain way, described by a path in the respective tree, then any world, corresponding to a node in that path, marks the corresponding action as forbidden (according to the intuition that forbidding a sequence means forbidding all the actions on that sequence). The third and last condition states that a successful realization of a forbidden action leads to a world in which a proposition defining a reparation holds.

The two conditions for a permitted action state respectively that any permitted action is possible, and that any step of such an action is not forbidden (although it may be executable in parallel with a forbidden action).

Properties of deontic notions in the system

Most of the basic axioms of DAL based on BA concerning permission and forbiddance are valid in the discussed system based on Kleene algebra[7]:

$$\mathbf{Perm}(\alpha \sqcup \beta) \equiv \mathbf{Perm}(\alpha) \wedge \mathbf{Perm}(\beta);$$

$$\mathbf{Forb}_{\mathscr{C}}(\alpha \sqcup \beta) \equiv \mathbf{Forb}_{\mathscr{C}}(\alpha) \wedge \mathbf{Forb}_{\mathscr{C}}(\beta);$$

$$\mathbf{Forb}_{\mathscr{C}}(0).$$

Moreover, formula:

$$\mathbf{Forb}_{\mathscr{C}}(\alpha) \to \mathbf{Forb}_{\mathscr{C}}(\alpha \sqcap \beta)$$

is also valid. However, unlike in those systems, permission and forbiddance are not symmetrical here. The following formulas are not valid:

$$\mathbf{Perm}(0);$$

$$\mathbf{Perm}(\alpha) \to \mathbf{Perm}(\alpha \sqcap \beta).$$

The non-validity of the former makes it possible for the following formula to be valid:

[7] Proofs of the facts concerning validity and non-validity of formulae stated here can be found in [25].

$$\mathbf{Perm}(\alpha) \to \neg\mathbf{Forb}_{\mathscr{C}}(\alpha).$$

Let us now apply the criteria that were used in [30] to compare various deontic action logics of permission and forbiddance based on boolean algebra, esp. Segerberg's system: closedness and treatment of 'doing nothing'. Although the set of basic action is finite, the absence of the classical complement makes it impossible to use the notion of atom from boolean algebra. Instead, the actions from $A^{\sqcap}$, which we called quasiatoms, can be used. The logic from [25] is closed neither for basic actions nor for quasiatoms. On the other hand 'doing nothing' is represented by action 1 which is quite different from the one considered in [30].

As for obligation the following formulas are valid:

$$\neg\mathbf{Obl}_{\mathscr{C}}(0);$$

$$\mathbf{Obl}_{\mathscr{C}}(1);$$

$$\mathbf{Obl}_{\mathscr{C}}(\alpha) \to \mathbf{Perm}(\alpha).$$

Moreover, the following formulas are not valid:

$$\mathbf{Obl}_{\mathscr{C}}(\alpha) \to \mathbf{Obl}_{\mathscr{C}}(\alpha \sqcap \beta);$$

$$\mathbf{Obl}_{\mathscr{C}}(\alpha \sqcap \beta) \to \mathbf{Obl}_{\mathscr{C}}(\alpha);$$

$$\mathbf{Obl}_{\mathscr{C}}(\alpha) \to \mathbf{Obl}_{\mathscr{C}}(\alpha \sqcup \beta);$$

$$\mathbf{Obl}_{\mathscr{C}}(\alpha \sqcup \beta) \to \mathbf{Obl}_{\mathscr{C}}(\alpha);$$

$$\mathbf{Obl}_{\mathscr{C}}(\alpha) \wedge \mathbf{Obl}_{\mathscr{C}}(\beta) \to \mathbf{Obl}_{\mathscr{C}}(\alpha \sqcap \beta).$$

The last formula, however, becomes valid if we add the following condition to the semantics of obligation:
there exists γ s.t. $T(\alpha \sqcap \gamma)$ is isomorphic to a simulating substructure of K w.r.t. $T(\alpha)$ and i.

The obligation modified in such a way is called in [25] a *natural obligation*.

For natural implication the following interesting formula is also valid:

$$\mathbf{Obl}_{\mathscr{C}_1}(\alpha) \wedge \mathbf{Obl}_{\mathscr{C}_2}(\beta) \to \mathbf{Obl}_{\mathscr{C}_1 \vee \mathscr{C}_2}(\alpha \sqcap \beta).$$

The way the definitions of obligation and prohibition are constructed guarantees that reparation is inevitable. Any possible execution of violating action by definition must end in a situation in which the deontic proposition describing a reparation holds. In particular for $\mathbf{Obl}_{\perp}(\alpha)$ there is no final path of $T(\overline{\alpha})$ strongly simulated by K. Intuitively, this means that it is impossible to violate absolute obligation and such an obligation can be understood as necessity. A similar fact holds for absolute forbiddance and consequently, it that can be interpreted as impossibility.

1.4.6 Conflicts between actions and specialized algebras

In the systems described above, we considered those action algebras generated by a finite set of basic actions. In the most straightforward situation all the combinations of basic actions are possible. However, it is not necessarily true. If actions a and b cannot be executed together, then their parallel combination is impossible, this can be expressed in symbols by the equation: $a \sqcap b = 0$. As an obvious example we can take actions: ‘turn left’ and ‘turn right’. Moreover, some actions, essentially available for an agent, may be impossible in some situations. For example, we can consider the action ‘turn left’ when there is no left turn available on the crossroads.

The above mentioned facts can be used to enrich the expressive power of deontic logic based on boolean (or Kleene) algebra. In [25], the notion of conflict often found in legal contracts is introduced as a relation imposing more structure into the algebra. It is defined as a symmetric and irreflexive relation over basic actions and symbolically represented by #. Its meaning is ensured by the following formula:

$$a\#b \rightarrow a \sqcap b = 0.$$

It can be further used in the deontic context to derive the following law:

$$\alpha\#\beta \rightarrow \neg(\mathbf{Obl}_{\mathscr{C}}(\alpha) \wedge \mathbf{Obl}_{\mathscr{C}}(\alpha)).$$

In [32], the possibility of defining multiple action algebras based on the same set of basic actions was used to formulate a strategy of building a system of norms. By that strategy, first, each situation in which an agent can find itself should be analyzed. The possible actions for all situations should be recognized and formulated in a boolean algebra. The deontic notions can be then introduced for each situation separately, defining what in each situation is permitted, forbidden and obligatory. Finally, actions can be collected from specific situations and used to formulate a general algebra of actions for agents. It is shown how to construct the characteristics of deontic notions for this algebra from their specification in specific situations.

1.5 Future Challenges

In section 1.3 we reviewed the logic defined by Segerberg, while in section 1.4 we have described several related logics that use a boolean algebra of actions and provide different formalizations of the deontic operators. In this section we discuss some further work about deontic action logic based on boolean algebra.

1.5.1 First-order deontic action logics

First, we review possible extensions of the logics described above aimed to embrace first-order reasoning. First-order deontic logics have been a topic of discussion since the beginning of deontic logic; for example, Hintikka [14] discusses the intuitive properties of first-order operators when combined with deontic operators; first-order operators are also explicit in the foundational work of Stig Kanger about ethical theory [20]. The main difficulty in deontic action logic to deal with first-order operators is the interplay between quantifiers and actions. In [8], the authors propose to introduce generalized boolean operators to deal with parameters, for example, consider the following term:

$$\bigsqcup_x a(x)$$

where a is an action letter. Roughly speaking, this operator is a non-deterministic execution of action a with some parameter x. For example, we may consider the following term:

$$\bigsqcup_x pay\_tax(x)$$

can be read as saying that some person pays its taxes. Some interesting questions arise when the first-order operators are introduced. For example, the proof of completeness in the propositional case relies on the fact that the underlying boolean algebra of terms (denoting actions) is atomic, and therefore the atoms in this algebra can be used to build a canonical model. It is not straightforward (at first sight) to preserve this property when the quantifiers are added; adding parameters to actions produces a boolean algebra of terms which is not atomic. The relationship between deontic operators and first-order predicates seems an interesting topic to investigate, for instance, it is not obvious at first sight which of this properties should be true:

- $\forall x : \mathbf{Perm}(\alpha(x)) \rightarrow \mathbf{Perm}(\bigsqcup_x \alpha(x))$.
- $\mathbf{Perm}(\bigsqcup_x \alpha(x)) \rightarrow \forall x : \mathbf{Perm}(\alpha(x))$,

and similar properties for weak permission and the existential operator. For example, it seems obvious that the first property should be true: *if all the persons are permitted to drink, then any chosen person will be allowed to drink*. Similarly, the second property also seems true: *if a person (selected in a non-deterministic way) is allowed to drink, then all the person are allowed to drink*. These properties are more complicated when obligation is involved, we refer the reader to the discussion in [20] about this properties. For example, in the logic propose by Kanger, we can write $Ax : O(Px)$, this is a quantification over actions; the intuitive meaning of this expression is: every action of type P is obligatory to be performed. In the same way, we can write: $O(Ax : Px)$ which must be read as: *it is obligatory that every act of type A is performed*. The formula $Ax : O(Px) \rightarrow O(Ax : Px)$ is discarded with intuitive examples of the style: *in some settings, everyone ought to pay fines, but it is not true in every deontically perfect world, that everyone should pay fines.* As explained in [8], reasoning about these logics can be very hard. Introducing generalized boolean operators, on the other hand, can allow us to obtain logics expressive enough to capture interesting problems. In a similar way, cylindric algebras seem to be another possible way of extending boolean algebra of actions to obtain a framework where elementary operations can be captured; from our point of view these topics deserve further investigation and discussion.

1.5.2 Boolean Algebras with Operators

Boolean algebras with operators are obtained by enriching boolean algebras with a collection of additional operators f_i which satisfy:

- are join preserving:

$$f_i(x_0,\dots,x_j \vee y_k,\dots,x_n) = f_i(x_0,\dots,x_j,\dots,x_n) \vee f_i(x_0,\dots,y_k,\dots,x_n)$$

- are normal for each argument: $f_i(\dots,0,\dots) = 0$.

These extra operators allow us to capture other intuitive combinators of actions. Many useful formalisms can be captured as BAO, for example: modal logics, relation algebras, relevance logics, geometries, etc. Between these algebras, relational algebras are those which are extension of boolean algebras and in addition they have the following operators:

- ; – composition of relations.
- $^{-1}$ – converse of relations.
- e – identity for composition.

These operators satisfy the following axioms:

- $(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z)$
- $x \sqcup y = y \sqcup x$
- $x = \overline{\overline{x} \sqcup y} \sqcup \overline{\overline{x} \sqcup \overline{y}}$
- $x;(y;z) = (x;y);z$
- $x;e = x$
- $(x \sqcup y);z = (x;z) \sqcup (y;z)$
- $(x^{-1})^{-1} = x$
- $(x;y)^{-1} = y^{-1};x^{-1}$
- $x^{-1};\overline{x;y} \sqcup y = \overline{y}$

All the axioms of boolean algebra can be deduced from this set of formulae. Relation algebras are very expressive; however, they are not representable and the axiomatic system shown above is not complete with respect to the calculus of relations (there do not exist finite axiomatizations of relation algebras); also the system is not decidable. If one intends to add operators such as ; or $^{-1}$, a correct way to start is looking at the theory of relation algebras []. It seems interesting to try to capture the meaning of the following predicates using algebraic methods:

$$\mathbf{Perm}(\alpha;\beta)$$

(which means that it is allowed to perform β after performing α), or:

$$\mathbf{Perm}(\alpha^{-1})$$

These kinds of operators have been discussed in the literature [6]; however, no algebraic methods are used by those authors; it seems an interesting trend of future research to investigate the interplay between deontic operators and these relational combinators. Another interesting algebras are the so-called *residuated boolean algebras* [], there exist residuated algebras that have finite axiomatization and that support the notion of atom, and therefore they provides an expressive framework where it is possible to express action properties.

1.6 Further Remarks

In this chapter we have reviewed those deontic action logics that are based on boolean algebra; this line of research was initiated by Segerberg, and continued by several authors []; the main characteristics of this approach is that deontic notions such as permission, prohibition and obligation can be captured using algebraic notions like ideals, filters, etc. However, one problematic issue of Segerberg' s logic is the lack of expressiveness to capture the closure principle of jurisprudence. We have introduced logics that use boolean atomic algebras to capture deontic operators; the main benefit of doing this is the possibility of using the atoms to state properties of the operators, in particular, this is important when capturing the closure principle. Future lines of research include the investigation of formalisms that allow one to introduce first-order reasoning and the use of boolean algebras with operators. We think that the main contribution of these formalisms is the possibility of studying the properties of deontic operators by means of well-known mathematical concepts like ideal, filters, etc. Furthermore, the use of algebraic tools seems to be a promising way of reasoning about more complicated action operators such as composition and iteration.

References

1. Anglberger, A.J.J.: Dynamic deontic logic and its paradoxes. Studia Logica **89**, 427–435 (2008)
2. Aquist, L.: Postulate sets and decision procedures for some systems of deontic logic. Theoria **29**, 154–175 (1963)
3. Belnap, N., Perloff, M., Xu, M.: Facing the Future: Agents and Choices in Our Indeterminist World. Oxford University Press (2001)
4. Blackburn, P., Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science 53 (2001)
5. Boole, G.: An Investigation on the Laws of Though, on which are founded the Mathematical Theories of Logic and Probability. Walton & Maberly (1854)

6. Broersen, J.: Modal action logics for reasoning about reactive systems. Ph.D. thesis, Vrije University (2003)
7. Castro, P.F., Maibaum, T.: Deontic action logic, atomic boolean algebra and fault-tolerance. Journal of Applied Logic **7**(4), 441–466 (2009)
8. Castro, P.F., Maibaum, T.: Towards a first-order deontic action logic. In: 20th International Workshop in Recent Trends in Algebraic Development Techniques, Lectures Notes in Computer Science. Springer (2010)
9. Fiadeiro, J.L., Maibaum, T.S.E.: Temporal reasoning over deontic specifications. J. Log. Comput. **1**, 357–395 (1991)
10. Fisher, M.: A three-valued calculus for deontic logic. Theoria **27**, 107–118 (1961)
11. Gargov, G., Passy, S.: A note on boolean logic. In: P.P.Petkov (ed.) Proceedings of the Heyting Summerschool. Plenum Press (1990)
12. Givant, S., Halmos, P.: Introduction to Boolean Algebras. Springer (2010)
13. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000)
14. Hintikka, J.: Quantifiers in deontic logic. In: Societas Scientiarum Fennica, Commentationes Humanarum Litterarum (1957)
15. Jonsson, B., Tarski, A.: Boolean algebras with operators i. Amer. J. Math. **73**, 891–939 (1951)
16. Jonsson, B., Tarski, A.: Boolean algebras with operators ii. Amer. J. Math. **74**, 127–162 (1952)
17. Kalinowski, J.: Theorie des propositions normativess. Studia Logica **1**, 147–182 (1953)
18. Kalinowski, J.: La logique des normes. Presses Universitaires de France (1972)
19. Kanger, S.: New foundations for ethical theory. Tech. rep., Stockholm University (1957)
20. Kanger, S.: New foundations for ethical theory. In: Deontic Logic: Introductory and Systematic Readings. Dordrecht (1971)
21. Maddux, R.: Relation Algebras. Elsevier Science (2006)
22. van der Meyden, R.: The dynamic logic of permission. J. Log. Comput. **6**(3), 465–479 (1996)
23. Meyer, J.: A different approach to deontic logic: Deontic logic viewed as variant of dynamic logic. Notre Dame Journal of Formal Logic **29**(1), 109–136 (1987)
24. Prisacariu, C.: Synchronous kleene algebra. The Journal of Logic and Algebraic Programming **78**, 608–635 (2009)
25. Prisacariu, C., Schneider, G.: A dynamic deontic logic for complex contracts. The Journal of Logic and Algebraic Programming **81**, to appear (2012)
26. Segerberg, K.: A deontic logic of action. pp. 269–282. Dordrecht, Reidel (1982)
27. Segerberg, K.: A topological logic of action. Studia Logica **43**(4), 415–419 (1984)
28. Segerberg, K.: Getting started: Beginnings in the logic of action. Studia Logica **51**, 347–378 (1992)
29. Sikorski, R.: Boolean Algebras. Springer-Verlag (1969)
30. Trypuz, R., Kulicki, P.: A systematics of deontic action logics based on boolean algebra. Logic and Logical Philosophy **18**, 263–279 (2009)
31. Trypuz, R., Kulicki, P.: Towards metalogical systematisation of deontic action logics based on boolean algebra. In: Proc. 10th International Conference Deontic Logic in Computer Science, *Lecture Notes in Computer Science*, vol. 6181. Springer (2010)
32. Trypuz, R., Kulicki, P.: A norm-giver meets deontic action logic. Logic and Logical Philosophy **20**, 59–72 (2011)
33. von Wright, G.H.: Deontic logic. Mind **LX**(237), 1–15 (1951)
34. von Wright, G.H.: Norm and Action: A Logical Inquiry. Routledge & Kegan Paul (1963)
35. von Wright, G.H.: Problems and prospects of deontie logic: A survey. In: Modern Logic - A Survey, pp. 399–423. Dordrecht, Reidel (1980)