GERT-JAN C. LOKHORST

Reasoning About Actions and Obligations in First-Order Logic*

Abstract. We describe a new way in which theories about the deontic status of actions can be represented in terms of the standard two-sorted first-order extensional predicate calculus. Some of the resulting formal theories are easy to implement in Prolog; one prototype implementation—R. M. Lee's deontic expert shell DX—is briefly described.

Key words: deontic logic, logic of action.

1. Introduction

We shall describe a new way in which theories about the deontic status of actions can be represented in terms of the standard two-sorted first-order extensional predicate calculus. This approach towards the formal analysis of such theories has two attractive features: first, it stands in the long and venerable 'anti-intensional' or 'anti-modal' tradition in the logic of action [11, 3, 7] and deontic logic [17, 15, 16]; and second, it leads to formal theories which are sometimes easy to implement in Prolog. Our analysis is preferable over previous proposals because it is both simpler and more comprehensive.

We shall first show how sentences about the deontic status of actions can be represented in the language of the first-order predicate calculus. We shall then give a brief description of Ronald M. Lee's deontic expert system DX which is based on these ideas.[1] We shall finally propose some extensions of the system. It will emerge that a considerable part of the currently popular 'modal' account of the deontic logic of action can be embedded in these—still purely first-order—extensions of the original system.

2. Basic Representational Issues

2.1. Facts

It is customary to distinguish between three types of facts: events, processes and states of affairs [18, ch. II.5]. The difference between these types of

*This research was partially supported by the ESPRIT III Basic Research Working Group No. 8319 MODELAGE.

[1]The historical sequence of events is just the reverse: our systems were originally inspired by Lee's DX.

Studia Logica **57**: 221–237, 1996.

facts is, roughly, this: events happen at a certain time, whereas processes and states of affairs have a certain duration; processes are 'dynamic', whereas states of affairs are 'static'. Events occur; processes go on; states of affairs obtain.

Facts can be *described* by means of *sentences* as well as *named* by means of *singular terms.*[2] The sentence 'Job is poor' is an example of a *description* of a fact; it can be represented by means of a formula of the form $F(a)$, where F is a monadic predicate symbol, standing for 'is poor', and a an individual constant, standing for 'Job'. 'Job's poverty', on the other hand, is a *name* of a fact; it can be represented by means of a singular term of the form $f(a)$, where f is a monadic function symbol, standing for 'the poverty of', and a an individual constant, standing for 'Job'.

2.2. Actions

Corresponding to the just-mentioned three types of facts, there are three types of actions: acts, activities and states of activity [18, ch. III.2]. Acts are events:[3] they occur at a certain time. Activities are processes: they go on over a certain period of time. States of activity are states of affairs: they obtain during a certain period.

Just like events, processes and states of affairs, acts, activities and states of activity can be represented in two ways: by means of sentences and by means of singular terms. 'God is bringing about Job's poverty' is an example of an action sentence. It can be represented by a formula of the form $\mathsf{Eff}(g, f(a))$, where the binary predicate symbol Eff stands for 'brings about', the individual constant g for 'God' and the singular term $f(a)$ for 'Job's poverty'. 'God's bringing about Job's poverty', on the other hand, is an example of an action term. It can be represented by a singular term of the form $\mathsf{eff}(g, f(a))$, where the binary function symbol eff stands for '—'s bringing about of $\cdots$', and the terms g and $f(a)$ have the same meaning as before.

Each action has an agent and an effect (result, outcome, post-condition), so all English action sentences and action terms can be represented by constructions of the forms $\mathsf{Eff}(x, y)$ and $\mathsf{eff}(x, y)$. The effect of an action may again be an action, so one may come across constructions such as $\mathsf{Eff}(x, \mathsf{eff}(y, f(z)))$ and $\mathsf{eff}(x, \mathsf{eff}(y, f(z)))$.

[2] This was stressed by Reichenbach [11, §48], whose own analysis was, however, different from ours, as will become clear below.

[3] See [3, p. 113] *contra* [18].

2.3. Deontic Status

Some events, processes and states of affairs (including actions, activities and states of activity) have deontic status: they are forbidden, obligatory, permitted, or waived (i.e., not obligatory). The deontic status of events, processes and states of affairs may be represented by means of predicate symbols whose arguments are singular terms. An example: 'God's bringing about Job's poverty is permitted' (or 'God is allowed to make Job poor') may be represented as $\mathsf{Perm}(\mathsf{eff}(g, f(a)))$, where the monadic predicate symbol Perm stands for 'is permitted'.

Von Wright seems to have been the first philosopher who analyzed deontic concepts by means of deontic predicate symbols [17]. He later abandoned this approach [18], but others have made it clear that it is still worth exploring [9, 14, 15, 16].

3. The Logical System DA

Let us now make these ideas more precise and explicit. The first deontic action logic which we shall present is basically the same as standard two-sorted first-order predicate logic except that some symbols are interpreted in special ways. Let us first give a brief recapitulation of standard two-sorted first-order logic with two basic sorts, 0 and 1.

3.1. The Language (LA)

The list of primitive symbols is as follows:

1. individual variables of sort 0: x, y, ...;
2. individual constants of sort 0: a, b, ...;
3. individual variables of sort 1: χ, ξ, ...;
4. individual constants of sort 1: α, β, ...;
5. for every $n \geq 1$, every string $\sigma \in \{0,1\}^\star$ of length n (where $\star$ is the Kleene star) and every sort $\tau \in \{0,1\}$, a denumerable sequence of n-ary function symbols of sort $\sigma \mapsto \tau$: $f_1^{\sigma\mapsto\tau}$, $f_2^{\sigma\mapsto\tau}$, ...;
6. for every $n \geq 1$ and every string $\sigma \in \{0,1\}^\star$ of length n, a denumerable sequence of n-ary predicate symbols of sort σ: F_1^σ, F_2^σ, ...;
7. connectives: $\neg$, $\wedge$;

8. quantifier: $\forall$;

9. punctuation symbols: ',', '(' , ')'.

The superscripts of the function symbols and predicate symbols will sometimes be omitted.

The notions 'term', 'atomic formula' and 'formula' are inductively defined as follows:

1. all individual constants and variables of sort σ are terms of sort σ;

2. if f is an n-ary function symbol of sort $\sigma_1 \cdots \sigma_n \mapsto \sigma_{n+1}$ and $t_1, \ldots, t_n$ are terms of sorts $\sigma_1, \ldots, \sigma_n$, respectively, then $f(t_1, \ldots, t_n)$ is a term of sort σ_{n+1};

3. if F is an n-ary predicate symbol of sort $\sigma_1 \cdots \sigma_n$ and $t_1, \ldots, t_n$ are terms of sorts $\sigma_1, \ldots, \sigma_n$, respectively, then $F(t_1, \ldots, t_n)$ is an atomic formula;

4. all atomic formulas are formulas;

5. if ϕ and ψ are formulas and v is a variable, then $\neg\phi$, $\phi \wedge \psi$ and $\forall v\phi$ are formulas.

$\vee$, $\rightarrow$, $\leftrightarrow$, $\bot$, $\top$ and $\exists$ are defined as usual. A *Horn clause* is a universally quantified formula of the form $(\phi_1 \wedge \cdots \wedge \phi_n) \rightarrow \phi_{n+1}$, where $\phi_1, \ldots, \phi_{n+1}$ are atomic formulas.

3.2. Formal Semantics

A model for DA is a structure

$$\mathfrak{M} = \langle D_0, D_1, V \rangle,$$

where D_0 and D_1 are non-empty sets and V is a function such that:

1. $V(t) \in D_\sigma$ for each term t of sort σ;

2. $V(f)$ is a mapping from $D_{\sigma_1} \times \cdots \times D_{\sigma_n}$ to $D_{\sigma_{n+1}}$ for each n-ary function symbol f of sort $\sigma_1 \cdots \sigma_n \mapsto \sigma_{n+1}$;

3. $V(F) \subseteq D_{\sigma_1} \times \cdots \times D_{\sigma_n}$ for each n-ary predicate symbol F of sort $\sigma_1 \cdots \sigma_n$.

$\mathfrak{M} \models \phi$ means that ϕ is true in $\mathfrak{M}$. This notion is defined as follows:

1. $\mathfrak{M} \models F(t_1, \ldots, t_n)$ iff $\langle V(t_1), \ldots, V(t_n)\rangle \in V(F)$;
2. $\mathfrak{M} \models \neg\phi$ iff $\mathfrak{M} \not\models \phi$;
3. $\mathfrak{M} \models \phi \wedge \psi$ iff $\mathfrak{M} \models \phi$ and $\mathfrak{M} \models \psi$;
4. $\mathfrak{M} \models \forall v\phi$ iff $\mathfrak{M}(d/v) \models \phi$ for all $d \in D_\sigma$, where v is a variable of sort σ, and where $\mathfrak{M}(d/v)$ is the model which is identical to $\mathfrak{M}$ except that $V(v) = d$.

$\Sigma, \mathfrak{M} \models \phi$ means that if $\mathfrak{M} \models \phi_i$ for all $\phi_i \in \Sigma$, then $\mathfrak{M} \models \phi$. $\Sigma \models \phi$ means that $\Sigma, \mathfrak{M} \models \phi$ for all models $\mathfrak{M}$.

3.3. Axiomatization

Axiom schemes:

(DA1) all truth-functional tautologies;

(DA2) $\forall v\phi(v) \rightarrow \phi(t)$, where t is free for v in ϕ, provided that v and t are of the same sort.

Rule schemes (where $\vdash \phi$ means that ϕ is a theorem):

(DAR1) if $\vdash \phi$ and $\vdash \phi \rightarrow \psi$, then $\vdash \psi$;

(DAR2) if $\vdash \phi \rightarrow \psi$ then $\vdash \phi \rightarrow \forall v\psi$, provided that v is not free in ϕ.

$\Sigma \vdash \phi$ means that ϕ is derivable from Σ, a notion which is defined as usual.

3.4. Soundness and Completeness

$\Sigma \vdash \phi \Leftrightarrow \Sigma \models \phi$. See, e.g., [4, ch. 8].

3.5. Informal Interpretation

Sort 0 will be regarded as the category of individual objects (including agents), sort 1 as the category of individual events, processes and states of affairs (including acts, activities and states of activity). All entities in the latter class will be called 'events' for the sake of brevity. Thus, individual variables of sorts 0 and 1 are individual object variables and individual event variables, repectively; individual constants of sorts 0 and 1 are individual object constants and individual event constants, respectively; and the sets D_0 and D_1 (in the formal semantics) are the sets of individual objects and individual events, respectively.

3.6. Designated Symbols

The following predicate symbols and function symbols are notated and interpreted in special ways.

Symbol	Meaning	Notation	Symbol	Meaning	Notation
F_1^1	is forbidden	Forb	$f_1^{1\mapsto 1}$	being forbidden	forb
F_2^1	is obligatory	Obl	$f_2^{1\mapsto 1}$	being obligatory	obl
F_3^1	is permitted	Perm	$f_3^{1\mapsto 1}$	being permitted	perm
F_4^1	is waived	Waiv	$f_4^{1\mapsto 1}$	being waived	waiv
F_1^{01}	brings about	Eff	$f_1^{01\mapsto 1}$	bringing about	eff

Note that, in all these cases, $f_i^{\sigma\mapsto 1}$ represents the *gerund* of the predicate expressed by F_i^σ. We shall adopt the following important interpretational convention:

$f_i^{\sigma\mapsto 1}$ represents the *gerund* of the predicate expressed by F_i^σ

3.7. Examples of Terms and Formulas

In the following examples, copier is a constant of sort 0 and empl a function symbol of sort $0 \mapsto 1$.

1. Eff$(a,$ empl(copier)) ['a brings about the employment of the copier', 'a uses the copier'];

2. eff$(a,$ empl(copier)) ['a's [bringing about the] employment of the copier'];

3. Perm(eff$(a,$ empl(copier))) ['a is allowed to use the copier'];

4. Perm(eff$(a,$ perm(eff$(b,$ empl(copier))))) ['a is permitted to permit b to use the copier'];

5. Eff$(a,$ eff$(b,\alpha))$ ['a makes b do α'].

Note that Forb, Obl, Perm, Waiv and Eff cannot occur within the scope of Forb, Obl, Perm, Waiv and Eff. Forb, Obl, Perm, Waiv and Eff are predicate symbols; only terms can occur in their range. These terms may, of course, contain function symbols, such as eff and perm in the last two examples.

3.8. Defined Operators

1. $[F_i^\sigma(t_1,\ldots,t_n)]^* \stackrel{\text{df}}{=} f_i^{\sigma\mapsto 1}(t_1,\ldots,t_n)$;[4]
2. $\mathrm{F}(\phi) \stackrel{\text{df}}{=} \mathsf{Forb}([\phi]^*)$;
3. $\mathrm{O}(\phi) \stackrel{\text{df}}{=} \mathsf{Obl}([\phi]^*)$;
4. $\mathrm{P}(\phi) \stackrel{\text{df}}{=} \mathsf{Perm}([\phi]^*)$;
5. $\mathrm{W}(\phi) \stackrel{\text{df}}{=} \mathsf{Waiv}([\phi]^*)$;
6. $\mathrm{E}(t,\phi) \stackrel{\text{df}}{=} \mathsf{Eff}(t,[\phi]^*)$.

An example:

$$\begin{aligned}\mathrm{P}(\mathrm{E}(a,\mathsf{Empl}(\mathsf{copier}))) &= \mathsf{Perm}([\mathrm{E}(a,\mathsf{Empl}(\mathsf{copier}))]^*)\\ &= \mathsf{Perm}([\mathsf{Eff}(a,[\mathsf{Empl}(\mathsf{copier})]^*)]^*)\\ &= \mathsf{Perm}(\mathsf{eff}(a,\mathsf{empl}(\mathsf{copier}))).\end{aligned}$$

Note that F, O, P, W and E may occur within the scope of F, O, P, W and E.

Von Wright abandoned the first-order approach to deontic logic because he could not imagine how nested prohibitions, obligations, permissions and waivers could ever be handled in it [18, Preface]. The just-presented definitions suggest that he abandoned hope too early. Nested deontic constructions can be made sense of after all.

3.9. DX: A Partial Implementation of DA

The logical system DA is purely first-order, so the standard techniques from the field of logic programming can be applied. The fact that DA is two-sorted is no obstacle to this [6]. Ronald M. Lee's deontic expert shell DX is a partial implementation (written in Prolog) of the Horn clause fragment of DA [5, 10, 12, 13]. When one takes a look at Lee's description of DX, it is not immediately apparent that this is the project he has carried out. He

[4]The $[\ ]^*$ notation is taken from Reichenbach [11, p. 269]. However, Reichenbach regarded $[F_i^\sigma(t_1,\ldots,t_n)]^*$ as a *primitive predicate symbol* (of sort 1, from our point of view), whereas we regard it as a *defined singular term* (of sort 1). Reichenbach would have written the latter term as $(\iota v)[F_i^\sigma(t_1,\ldots,t_n)]^*(v)$, where v is a variable of sort 1 (in our terminology). An example may make this clearer. Let $F_i^0(x)$ stand for 'George VI is crowned'. We represent 'the coronation of George VI' as $[F_i^0(x)]^*$, which is by definition equivalent with $f_i^{0\mapsto 1}(x)$, whereas Reichenbach (ibid.) represented it as $(\iota v)[F_i^0(x)]^*(v)$.

gives the following Backus-Naur definition of his language, which we shall call LX [5, p. 11]:

```
<rule> ::= <condition>
<rule> ::= if <conditions> then <condition>
<rule> ::= <condition> if <conditions>
<conditions> ::= <condition>
<conditions> ::= <condition> and <conditions>
<condition> ::= <predicate>
<condition> ::= forbid(<action>)
<condition> ::= oblig(<action>)
<condition> ::= permit(<action>)
<condition> ::= waiv(<action>)
<action> ::= <agent>:<condition>
```

Lee's terminology is different from ours. Things fall into place as soon as the following translation table is used.

LX	LA
`forbid`	F
`oblig`	O
`permit`	P
`waiv`	W
: (infix)	E (prefix)
`<predicate>`	atomic formula
`<condition>`	atomic formula or formula of one of the following forms: $F(\phi)$, $O(\phi)$, $P(\phi)$, $W(\phi)$, where ϕ is an `<action>`
`<action>`	formula of the form $E(t, \phi)$, where ϕ is a `<condition>`
`<conditions>`	conjunction of `<condition>`s and `<action>`s
`<rule>`	Horn clause

The following formula is an example of a rule in Lee's sense:

`permit(X: permit(Y: use-copier)) if chair(X).`

This corresponds to the following formula of LA:

$$\forall x \forall y \ (\mathsf{Chair}(x) \rightarrow \mathrm{P}(\mathrm{E}(x, \mathrm{P}(\mathrm{E}(y, \mathsf{Empl}(\mathsf{copier})))))).$$

It will be clear that all formulas of LX can be translated into LA. The converse does not hold:

- LA-formulas of the forms $F(\phi)$, $O(\phi)$, $P(\phi)$ and $W(\phi)$ cannot be translated into LX unless ϕ is an action sentence, i.e., unless ϕ is of the form $E(t, \phi)$ (this is Lee's way of doing justice to the '*Tunsollen* rather than *Seinsollen*' thesis from classical ethics);

- LA-formulas of the form $\mathrm{E}(t, \phi)$ cannot be translated into LX if ϕ is itself an action sentence, i.e., if ϕ is of the form $\mathrm{E}(t', \psi)$.

These differences between LX and LA disappear if the last five lines of Lee's definition are replaced by the following ones:

```
<condition> ::= forbid(<condition>)
<condition> ::= oblig(<condition>)
<condition> ::= permit(<condition>)
<condition> ::= waiv(<condition>)
<condition> ::= <agent>:<condition>
```

4. The Logical System DB

4.1. Limitations of DA

DA has a major shortcoming: complex actions and obligations cannot be represented in it.[5] Some examples:

1. a's doing non-α;
2. a's doing α or β;
3. a's not doing α is obligatory (it is obligatory that a does not do α);
4. a's doing β upon doing α is obligatory (it is obligatory that if a does α, he does β).

The solution is simple: following Segerberg [15], we enrich the language with Boolean operators.[6]

4.2. The Language (LB)

LB is the same as LA, except that:

1. The following special symbols are added:
   (a) two designated individual constants of sort 1: **0** and **1**; the former represents the empty (impossible) event, the latter the universal (necessary) event;

[5] Reichenbach's work had the same shortcoming; he did not pay attention to complex events.

[6] There are several differences between our account and Segerberg's. (1) Segerberg considered only one sort of singular terms, namely event-terms. (2) He did not consider quantification over events, although he did use open formulas with event-variables which were implicitly regarded as being universally quantified. (3) Forb and Perm were taken as the primitive deontic predicates.

(b) a designated function symbol of sort $1 \mapsto 1$: $^-$; this symbol is read as 'not', 'non', or 'the complement of';

(c) two designated function symbols of sort $11 \mapsto 1$: $\sqcap$ and $\sqcup$; these symbols are read as 'and' and 'or', or as 'the intersection of' and 'the union of', respectively;

(d) a designated predicate symbol of sort 11: '='; this symbol is read as 'is identical with'.

2. Obl and obl are the only primitive deontic symbols; the other deontic symbols are defined as follows:

(a) $\mathsf{Forb}(t) \stackrel{\text{df}}{=} \mathsf{Obl}(\overline{t})$;

(b) $\mathsf{Perm}(t) \stackrel{\text{df}}{=} \neg\mathsf{Forb}(t)$;

(c) $\mathsf{Waiv}(t) \stackrel{\text{df}}{=} \neg\mathsf{Obl}(t)$;

(d) $\mathsf{forb}(t) \stackrel{\text{df}}{=} \mathsf{obl}(\overline{t})$;

(e) $\mathsf{perm}(t) \stackrel{\text{df}}{=} \overline{\mathsf{forb}(t)}$;

(f) $\mathsf{waiv}(t) \stackrel{\text{df}}{=} \overline{\mathsf{obl}(t)}$.

Note that only the second and third definitions are stateable in LA.

Some examples of terms and formulas (cf. §4.1.):

1. $\mathsf{eff}(a, \overline{\alpha})$ ['a's doing non-α'];

2. $\mathsf{eff}(a, \alpha \sqcup \beta)$ ['a's doing α or β'];

3. $\mathsf{Obl}(\overline{\mathsf{eff}(a, \alpha)})$ ['a's not doing α is obligatory'];

4. $\mathsf{Obl}(\overline{\mathsf{eff}(a, \alpha)} \sqcup \mathsf{eff}(a, \beta))$ ['a's doing β upon doing α is obligatory'].

4.3. Formal Semantics

A model for DB is a structure

$$\mathfrak{M} = \langle D_0, \mathcal{B}, V \rangle,$$

where D_0 is a non-empty set, $\mathcal{B} = \langle D_1, \underline{0}, \underline{1}, -, \cap, \cup \rangle$ is a Boolean algebra, and V is a function which satisfies the following conditions in addition to those which were already mentioned in the definition of the DA-models:

1. $V(\mathbf{0}) = \underline{0}$;

2. $V(\mathbf{1}) = \underline{1}$;
3. $V(\overline{t}) = -V(t)$;
4. $V(t_1 \sqcap t_2) = V(t_1) \cap V(t_2)$;
5. $V(t_1 \sqcup t_2) = V(t_1) \cup V(t_2)$;
6. $V(=) = \{\langle d, d\rangle : d \in D_1\}$.

Note that D_1 may be regarded as the power-set of the set D_1 in our previous models. V may accordingly be regarded as an assignment of *sets* of events (rather than single events) to event-terms.

4.4. Axiomatization

The following axiom schemes are to be added to the axiom schemes of DA:

(DB1) $\chi = \chi$;

(DB2) $\chi = \xi \rightarrow (\phi \leftrightarrow \phi(\xi/\chi))$, where $\phi(\xi/\chi)$ is a formula which arises from ϕ by replacing some free occurrences of χ by ξ and ξ is free for the occurrences of χ that it replaces;

(DB3) $\chi \sqcap \xi = \xi \sqcap \chi, \chi \sqcup \xi = \xi \sqcup \chi$;

(DB4) $\chi \sqcap (\xi \sqcup \zeta) = (\chi \sqcap \xi) \sqcup (\chi \sqcap \zeta), \chi \sqcup (\xi \sqcap \zeta) = (\chi \sqcup \xi) \sqcap (\chi \sqcup \zeta)$;

(DB5) $\chi \sqcap \mathbf{1} = \chi, \chi \sqcup \mathbf{0} = \chi$;

(DB6) $\chi \sqcap \overline{\chi} = \mathbf{0}, \chi \sqcup \overline{\chi} = \mathbf{1}$;

(DB7) $\mathbf{0} \neq \mathbf{1}$.

The rule schemes are as before.

4.5. Soundness and Completeness

$\Sigma \vdash \phi \Leftrightarrow \Sigma \models \phi$. Proof: similar to the proof in [15].

4.6. Implementability

We have not yet studied the implementability of (useful fragments of) DB. The new axiom schemes give rise to various complications, depending as they do on both the identity symbol [6, §14] and the complementation operator, which has the same computationally troublesome properties as classical logical negation.

4.7. Defined Operators

The following definitions are to be added to those in §3.8.:

1. $[\neg\phi]^* \stackrel{\text{df}}{=} \overline{[\phi]^*}$;

2. $[\phi \wedge \psi]^* \stackrel{\text{df}}{=} [\phi]^* \sqcap [\psi]^*$.

Note that $[\forall x\phi]^*$ is not defined. We shall also assume that $[t_1 = t_2]^*$ is not defined. Some examples of formulas:

1. $\mathrm{O}(\neg\mathrm{E}(a, \phi))$ ['it is obligatory that a does not do ϕ'];

2. $\mathrm{O}(\mathrm{E}(a, \phi) \rightarrow \mathrm{E}(a, \psi))$ ['it is obligatory that if a brings it about that ϕ, then he brings it about that ψ'].

4.8. Derived Rules of Inference

(DBR1) $\vdash_{\mathrm{PC}} \phi \leftrightarrow \psi \Rightarrow \vdash_{\mathrm{DB}} [\phi]^* = [\psi]^*$.

Here $\vdash_{\mathrm{PC}} \phi$ means that ϕ is a theorem of the propositional calculus. This rule scheme is a consequence of the correspondence between Boolean algebra and propositional logic; see, e.g., [8, §3.7]. Since ϕ and ψ contain no quantifiers and no occurrences of '=' ($[\phi]^*$ and $[\psi]^*$ would not be defined otherwise), and we clearly have $\vdash_{\mathrm{DB}} \phi \leftrightarrow \psi \Rightarrow \vdash_{\mathrm{PC}} \phi \leftrightarrow \psi$, provided that ϕ and ψ contain no quantifiers and no occurrences of '=', we may also, more simply, write:

(DBR2) $\vdash \phi \leftrightarrow \psi \Rightarrow \vdash [\phi]^* = [\psi]^*$.

The following rule schemes are derivable from the latter scheme:

(DBR3) $\vdash \phi \leftrightarrow \psi \Rightarrow \vdash \mathrm{O}(\phi) \leftrightarrow \mathrm{O}(\psi)$;

(DBR4) $\vdash \phi \leftrightarrow \psi \Rightarrow \vdash \mathrm{E}(x, \phi) \leftrightarrow \mathrm{E}(x, \psi)$.

It is to be noted that $(\phi \leftrightarrow \psi) \rightarrow [\phi]^* = [\psi]^*$ is not a theorem. This has an important consequence: Davidson's celebrated criticism [3, pp. 117–118] of Reichenbach's proposals [11, §48] is completely unjustified (cf. §5.4. below).

4.9. Expressive Limitations

A (partially) *de dicto* obligation such as 'everybody ought to love somebody someday' (free after Dean Martin) cannot be represented in LB. This sentence is representable as

$$\forall x \mathrm{O}(\exists y \exists d \mathsf{Loves}(x, y, d))$$

in standard deontic logic [1]. The latter formula does, however, not belong to LB. It remains to be seen whether this is a serious limitation. Suppose, for example, that there is only a finite number of people $h_1, \ldots, h_n$ and that each man or woman h_i has only a finite number of days $\star(i), \ldots, \dagger(i)$ on which he or she can love somebody. (Neither assumption is unrealistic.) The sentence can then be represented by the following LB-formula:

$$\bigwedge_{1 \leq i \leq n} \mathrm{O} \Big(\bigvee_{1 \leq k \leq n} \bigvee_{\star(i) \leq d \leq \dagger(i)} \mathsf{Loves}(h_i, h_k, d) \Big).$$

5. From DB to Standard Deontic Logic and Beyond

5.1. System DC

DC has the following axiom schemes in addition to those of DB:

(DC1) $\mathsf{Obl}(\mathbf{1})$;

(DC2) $\mathsf{Obl}(\chi \sqcap \xi) \leftrightarrow (\mathsf{Obl}(\chi) \wedge \mathsf{Obl}(\xi))$;

(DC3) $\mathsf{Eff}(x, \mathbf{1})$;

(DC4) $\mathsf{Eff}(x, \chi \sqcap \xi) \leftrightarrow (\mathsf{Eff}(x, \chi) \wedge \mathsf{Eff}(x, \xi))$;

(DC5) $\mathsf{obl}(\mathbf{1}) = \mathbf{1}$;

(DC6) $\mathsf{obl}(\chi \sqcap \xi) = \mathsf{obl}(\chi) \sqcap \mathsf{obl}(\xi)$;

(DC7) $\mathsf{eff}(x, \mathbf{1}) = \mathbf{1}$;

(DC8) $\mathsf{eff}(x, \chi \sqcap \xi) = \mathsf{eff}(x, \chi) \sqcap \mathsf{eff}(x, \xi)$.

The corresponding semantic conditions are straightforward. Note that (DC1)–(DC4) correspond to the condition that $V(\mathsf{Obl})$ and $\{e : \langle d, e \rangle \in V(\mathsf{Eff})\}$ are *filters* on $\mathcal{B}$.

The following formulas are derivable from (DC1)–(DC4):

1. $\mathrm{O}(\top)$;

2. $\mathrm{O}(\phi \wedge \psi) \leftrightarrow (\mathrm{O}(\phi) \wedge \mathrm{O}(\psi))$;

3. $\mathrm{E}(x, \top)$;

4. $\mathrm{E}(x, \phi \wedge \psi) \leftrightarrow (\mathrm{E}(x, \phi) \wedge \mathrm{E}(x, \psi))$.

Axioms (DC5)–(DC8) ensure that DC is closed under (DBR1) and hence under (DBR3) and (DBR4).

A sentential operator $\Box$ is known as a *normal modal operator* in system Σ iff the following principles hold:

1. $\vdash_\Sigma \Box(\top)$;

2. $\vdash_\Sigma \Box(\phi \wedge \psi) \leftrightarrow (\Box(\phi) \wedge \Box(\psi))$;

3. $\vdash_\Sigma \phi \leftrightarrow \psi \Rightarrow \vdash_\Sigma \Box(\phi) \leftrightarrow \Box(\psi)$.

It will be clear that O and E are normal modal operators in DC, just like O in standard deontic logic [1] and Δ ('sees to it that', 'brings it about that') in Chellas's logic of action [2].

5.2. System DD

One feature of DC is unsatisfactory: there is no clear logical relationship between ϕ and $[\phi]^*$. We remove this defect with the help of a predicate symbol Occ of sort 1. $\mathsf{Occ}(\alpha)$ is read as 'α occurs' (in case α is an event), 'α goes on' (in case α is a process) or 'α obtains' (in case α is a state). occ is the corresponding function symbol of sort $1 \mapsto 1$. System DD has the following axiom schemes in addition to those of DC:

(DD1) $\phi \leftrightarrow \mathsf{Occ}([\phi]^*)$;[7]

(DD2) $\chi = \mathsf{occ}(\chi)$.

The corresponding semantic conditions are straightforward. Axiom (DD2) ensures that DD is closed under (DBR1) and hence under (DBR3) and (DBR4).

5.3. System DE

Standard deontic logic (SDL) has the following axiom scheme:

(D) $\neg \mathrm{O}(\bot)$.

Furthermore, most authors (e.g., Chellas [2]) adopt the following axiom scheme for the action operator E:

[7] (DD1) is similar to Reichenbach's $\phi \leftrightarrow (\exists v)[\phi]^*(v)$ [11, p. 271 (9)]; cf. note 4.

(E) $\mathrm{E}(x, \phi) \rightarrow \phi$.

These formulas become derivable if the following axiom schemes are added to the axiom schemes of DD:

(DE1) $\neg\mathsf{Obl}(\mathbf{0})$;
(DE2) $\mathsf{Eff}(x, \chi) \rightarrow \mathsf{Occ}(\chi)$;
(DE3) $\mathsf{obl}(\mathbf{0}) = \mathbf{0}$;
(DE4) $\mathsf{eff}(x, \chi) \sqsubseteq \chi$.

The corresponding semantic conditions are straightforward. Note that (DE1) corresponds to the condition that $V(\mathsf{Obl})$ is a *proper* filter on $\mathcal{B}$. Axioms (DE3) and (DE4) ensure that DE is closed under (DBR1) and hence under (DBR3) and (DBR4).

The following metatheorems can now be proven:

(*) $\vdash_{\mathrm{SDL}} \phi$ iff $\vdash_{\mathrm{DE}} \phi$, provided that ϕ belongs to the languages of both DE and quantified SDL;

(**) $\vdash_{\mathrm{CLA}} \phi$ iff $\vdash_{\mathrm{DE}} \phi$, provided that ϕ belongs to the languages of both DE and quantified CLA, where CLA is Chellas's logic of action.

In other words, we have now covered the whole route from DX towards the standard modal account of deontic logic and the logic of action.

5.4. System DF

We may go even further: if we add the axiom scheme

$$\text{(DF)} \qquad (F_i^\sigma(x_1, \ldots, x_n) \leftrightarrow F_k^\sigma(x_1, \ldots, x_n)) \rightarrow f_i^{\sigma \mapsto 1}(x_1, \ldots, x_n) = f_k^{\sigma \mapsto 1}(x_1, \ldots, x_n)$$

to DE in order to obtain system DF, we may prove such theorems as the following ones:

1. $(\phi \leftrightarrow \psi) \rightarrow [\phi]^* = [\psi]^*$ (proof: by induction);
2. $(\phi \leftrightarrow \psi) \rightarrow (\mathrm{O}(\phi) \leftrightarrow \mathrm{O}(\psi))$;
3. $(\phi \leftrightarrow \psi) \rightarrow (\mathrm{E}(x, \phi) \leftrightarrow \mathrm{E}(x, \psi))$.

In other words, we get a purely extensional deontic logic of action. (Note, however, that O and E are, though extensional, definitely not truth-functional.) Since nobody would want such an excessively strong system (Davidson's criticism [3, pp. 117–118] of [11, §48] would apply to it), we mention DF only for the sake of curiosity.

5.5. Implementability

The remarks made in §4.6. apply here as well. Observations (*) and (**) suggest that DE is in the same boat as *de re* quantified SDL and CLA as far as implementational issues are concerned. So the implementational advantages which we seemed to have in the case of DA are quickly lost once the latter system is made more 'orthodox'.

6. Conclusion

Our enterprise started off as an investigation into the logical foundations of Lee's expert system DX. Although DX is rooted in the tradition of logic programming, its logical import was unclear at first sight. Uncovering the logical content of DX was not only an interesting puzzle: since DX is typical of much work in AI, our results may well be applicable in a wider domain.

As soon as the logical content of DX was clarified, DX turned out to be inadequate. Systems DB–DE demonstrate what has to be added in order to let DA/DX qualify as a full-fledged deontic logic of action. It is conceivable that similar expert systems in AI will have to be extended in analogous ways.

We think that it is important that further work is carried out along the lines we have sketched. Since there is a growing need for deontic expert systems, such work is not only of academic interest: Society is waiting for us.

References

[1] AQVIST, L., 1984, 'Deontic logic', In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic, Vol. II: Extensions of Classical Logic,* 605–714, Reidel, Dordrecht.

[2] CHELLAS, B. F., 1992, 'Time and modality in the logic of agency', *Studia Logica* **51**, 485–517.

[3] DAVIDSON, D., 1980, *Essays on Actions and Events,* chapter The Logical Form of Action Sentences (1967), 105–122. Clarendon Press, Oxford.

[4] HUGHES, G. E., and M. J. CRESSWELL, 1968, *An Introduction to Modal Logic,* Methuen, London.

[5] LEE, R. M., 1992, *DX: A deontic expert system. Working Paper* 1992.10.01, Erasmus University Research Institute for Decision and Information Systems (EURIDIS), Rotterdam.

[6] LLOYD, J. W., 1987, *Foundations of Logic Programming,* second edition, Springer-Verlag, Berlin.

[7] MARTIN, R. M., 1981, *Logico-Linguistic Papers,* chapter On the Analysis of Action Sentences, 155–169. Foris, Dordrecht.

[8] MENDELSON, E., 1970, *Boolean Algebra and Switching Circuits,* McGraw-Hill, New York.

[9] MEYER, J. J. CH., 1988, 'A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic', *Notre Dame Journal of Formal Logic* **29,** 109–136.

[10] ONG, K. L., and R. M. LEE, 1993, *A formal model for maintaining consistency of evolving bureaucratic policies: A logical and abductive approach. Research Monograph* 1993.01.01, Erasmus University Research Institute for Decision and Information Systems (EURIDIS), Rotterdam.

[11] REICHENBACH, H., 1947, *Elements of Symbolic Logic,* Macmillan, New York.

[12] RYU, Y. U., and R. M. LEE, 1992, *Defeasible deontic reasoning and its applications to normative systems. Working Paper* 1992.07.01, Erasmus University Research Institute for Decision and Information Systems (EURIDIS), Rotterdam.

[13] RYU, Y. U., and R. M. LEE, 1992, *A formal representation of normative systems. Research Monograph* 1992.08.01, Erasmus University Research Institute for Decision and Information Systems (EURIDIS), Rotterdam.

[14] ROYAKKERS, L. M. M., 1993, *Predikatieve deontische logica. Master's thesis,* Technische Universiteit, Eindhoven.

[15] SEGERBERG, K., 1982, 'A deontic logic of action', *Studia Logica* **41,** 269–282.

[16] SEGERBERG, K., 1984, 'A topological logic of action', *Studia Logica* **43,** 415–419.

[17] VON WRIGHT, G. H., 1951, 'Deontic logic', *Mind* **60,** 1–15.

[18] VON WRIGHT, G. H., 1971, *Norm and Action: A Logical Enquiry,* Routledge and Kegan Paul, London.

ERASMUS UNIVERSITY
RESEARCH INSTITUTE FOR DECISION
AND INFORMATION SYSTEMS (EURIDIS)
AND DEPARTMENT OF PHILOSOPHY,
ERASMUS UNIVERSITY, P.O. BOX 1738,
NL–3000 DR ROTTERDAM,
THE NETHERLANDS.
G.Lokhorst@euridis.fbk.eur.nl
Lokhorst@filint.fwb.eur.nl