Deontic Concepts for Controlled Natural Languages

Adam Wyner

Department of Computer Science
University of Liverpool
Liverpool, United Kingdom

2nd Workshop on Controlled Natural Languages
Marittemo, Italy

September 13, 2010

Introduction

- Deontic concepts - what are they and why bother?
- If they are so great, why aren't we using them?
- What do I suggest? (However, this is more a request for collaboration than an outline of a solution in natural language).
- What to do next?

Based on [2]

A 'high level' specification of issues on and suggestions for introducing deontic concepts into CNLs – linguistic, logical, and computational.

Deontic Concepts - What are they and why bother?

- Concepts of *obligation*, *permission*, and *prohibition*
- Express what an individual *ought* to do, *may* do, or *should not do*.
- *Deontic specifications* - ascribe a property to an action that an individual or a collective of individuals performs.
- The additional property is whether the action executed *violates* of *fulfils* the deontic specification.
- Help the agent to *guide* his/her behaviour in the sense that the agent *prospectively* considers the consequences of his actions relative to some deontic ascription to an action.
- Widely used in a range of human endeavours where there is some notion of regulation.

Some Simple Examples

There is a control board in which an agent Bill and a control board. There are two toggles, left and right; each toggle can be moved up or down, left or right. In total, there are eight possible actions.

- Bill is obligated to move the left toggle up.
- Bill is forbidden to move the right toggle left.
- Bill is permitted to move the right toggle down.

Some Simple Examples

The actions can be performed sequentially or simultaneously.

- Bill is obligated to move the left toggle up.
- If Bill violates his obligation to move the left toggle up, then Bill is obligated to move the right toggle right.
- If Bill fulfils his obligation to move the left toggle up, then Bill is obligated to move the right toggle left.

Bill reasons about what to do relative to his obligations and the consequences of his actions. For the obligation to be *effective* in guiding behaviour, there should be *consequences* which follow from the fulfilment or violation of the obligation - concrete punishments or rewards. Bill's reasoning is prospective with respect to what rewards to accrue or punishments to avoid. Notice that one obligation occurs in the context of the violation (fulfilment) of another obligation; actions and obligations trigger 'contract' update (The so-called CTD Paradox).

Reasoning with *Deviant* Behaviour rather than Ruling it Out

- A central goal of deontic reasoning is to enable reasoning with deviant behaviour rather than ruling it out altogether (so, system constraints are not sufficient).
- Deviant behaviour occurs; do not want a system 'lock' or 'existential void'; violation is not the same as inconsistency.

OK, so why aren't we using them in CNLs?

- Long, deep, complex literature (see a review in [1]).
- Deontic logic is a branch of modal logic, like *alethic* logic (necessity and possiblity), *epistemic* logic (belief), and *dynamic* logic (context change).
- Lots of axiomatic presentations of the operators, but these give rise to lots of "problems" and "paradoxes" of reasoning.
- Modal logics generally appear as sentential/propositional operators, which have not (yet) been well-addressed in CNLs.
- Many different approaches and little consensus – state-wise, state-changing, defeasible, event calculus....
- A range of *linguistic* issues (which have not been given much attention in general).

Obviously, I am **not** going to review all of this (see [2]).

Some Linguistic Issues

- Objective: show that linguistic analyis is relevant to the correct semantic interpretation; indicate some syntactic and semantic issues that might be taken into consideration in formalising a CNL. But, introduces lots of complications.
- The deontic operators are viewed as sentence operators like the alethic modals *necessity* and *possibility*, the tense operators *Past* and *Future*, or context-setting adverbs *Stupidly* and *Botanically*.
- Bill is obligated to move the left toggle up.
- Where **OB** is the operator *obligated*, and the sentence is *Bill moves the left toggle up*, the logical form is (in a sketch): **OB** [Bill moves the left toggle up]

Some Linguistic Issues

- Question: Are their different interpretations of the auxiliaries, and if so, how do we tell them apart?
- Question: What syntactic and semantic constraints appear; that is, can **OB** combine with *any* proposition or are there selection restrictions or syntactic/semantic interactions (e.g. scope over conditional or relative clauses)?

Different Interpretations of Auxiliaries

Different sentence frames (pick one for a CNL and stick to it?)

a. *Obligatorily*, Bill visits Jill.
b. It is *obligatory* that Bill visits Jill.
c. Bill is *obligated* to visit Jill.
d. Bill is *obliged* to visit Jill.
e. It *ought* to be that Bill visits Jill.
f. Bill *ought* to visit Jill.
g. Bill *must* visit Jill.

Ought and *must* are ambiguous between **OB** (root) and **OU** (epistemic). They have similarities (deviance from norms is marked and modality). Key differences are that the **OB** interpretation requires an agent, an event (see stative obligations), and give rise to a violation; the **OU** interpretation does not; *OU* quantifies over possible worlds; **OB** is a relationship between worlds.
How would a CNL enforce the selection restrictions?

Different Interpretations of Auxiliaries - Evidence

1a. # It is obliged to rain. (Pleonastic subjects)

1b. It ought to rain.

2a. # The stone is obliged to weigh 10 kilos. (Non-agentive subjects)

2b. The stone ought to weigh 10 kilos.

3a. Bill is obliged to X visit Jill. (X must be controlled by the matrix subject)

3b. Bill ought to X visit Jill.
Bill is the bearer of the obligation and the agent of visiting.

4a. # Bill is obliged to realize the answer. (Selection restrictions on verbs - Psych verb example)

4b. Bill ought to realize the answer.

Operator Scope and Conditionals

a. It is obligatory that if you leave tomorrow, then we drink tonight.

b. If it is obligatory that you leave tomorrow, then it is obligatory that we drink tonight.

c. If you leave tomorrow, then it is obligatory that we drink tonight.

Operator Scope and Conditionals

- In modal logic, axioms such as Axiom-K are considered: if $\Box$ [P $\rightarrow$ Q], then [$\Box$P $\rightarrow$ $\Box$Q]
- These are important for the semantic analysis of the operator.
- Suppose **OB** is a modal operator like $\Box$ in many respects. Where can it appear and does it abide by Axiom-K (or the other axioms of modal logic)?
- Considering linguistic analyses, I concluded that forms such as **OB**[P $\rightarrow$ Q], [**OB**P $\rightarrow$ **OB**Q], and [P $\rightarrow$ **OB**Q] are all grammatical and interpretable, but are not logically related. There seems to be a default interpretation of **OB**[P $\rightarrow$ Q] as [P $\rightarrow$ **OB**Q]. There are a range of issues about [**OB**P $\rightarrow$ **OB**Q].

Example of a Paradox and Mention of Linguistic Solution - the Good Samaritan Paradox

a. It is obligatory that Bill help Phil, who has had an accident.

b. Therefore, it is obligatory that Phil has had an accident.

Intuitively, [a.] can be true, but [b.] false. However, (at least some/most) analyses in deontic logic have it that [a.] entails [b.].

a. Bill helps Phil and Phil has had an accident.

b. [It is obligatory that (Bill helps Phil and Phil has had an accident)] entails [It is obligatory that Bill helps Phil] and [It is obligatory that Phil has had an accident] (OB-RM).

We show that using a linguistic distinction between *restrictive* and *non-restrictive relative clauses* (NRC), the paradox is avoided; NRCs are known to be outside the scope of sentential operators.

Two Other Topics - Stative Obligations

- The yard must be clean.

Would seem to violate claims that there are agents and actions.

I argue that the 'underlying' meaning here is along the lines of:

- Someone must maintain the yard clean.

In other words, whenever the yard gets dirty (leaves, dust, trash, etc), someone is obligated to clean it.

Two Other Topics - Deadlines

It is widely claimed that obligations intrinsically have deadlines:

- Bills must be paid within 30 days after receiving the invoice. Otherwise, a fine must be paid.

But, this is only relevant for *telic* obligations, not *atelic* obligations:

a. John is obligated to build a house in a month. (Telic)

b. John is obligated to build houses for a month. (Atelic)

What are the *violation* conditions for atelic actions (or states, by the same token)?

So Far....

- Shown that there are a range of syntactic and semantic considerations that might be taken into consideration in providing a fully developed analysis.
- A various and large mix of (complex and interacting) issues.
- There are (many) others and few fully examined.
- For CNLs, we should highly constrain where and how the operators are used.

Aspects of an Analysis

- [2] focusses on a dynamic action logic and a core issue bearing on the analysis of the CTD paradox and compositionality of the violation and fulfilment markers.
- A *reductionist* analysis of the deontic specifications.
- Not an axiomatic analysis, but it allows constraints; a flexible and 'open' analysis that can be adapted.
- Action negation. *Bill is obligated to leave the house* means that Bill fulfils the obligation when he leaves the house, violates it if he doesn't leave the house, but says nothing about other actions he may otherwise perform. Action antonyms vs. set complement.

Caveat: the analysis (and its implementation) is abstract, with actions as state-changing functions and agents as individuals. There is much work to do to make it work for natural language expressions.

Intuitive Reductionist Analysis (Reanalysis of the CTD)

a. It is obligatory that Bill visits Jill.

b. If Bill has fulfilled his obligation to visit Jill, it is obligatory that he leaves his visiting card with her.

c. If Bill has violated his obligation to visit Jill, it is obligatory that he sends a letter of regret to her.

d. If it is obligatory that Bill visits Jill and Bill visits Jill, then Bill has fulfilled his obligation to visit Jill.

e. If it is obligatory that Bill visits Jill and Bill does not visit Jill, then Bill has violated his obligation to visit Jill.

f. Bill does not visit Jill.

Explicit introduction of a violation (fulfilment) expression for reasoning and representation. The violation (fulfilment) expressions *are as* **fine-grained** *and* **compositional** *as the expression the deontic operator applies to*. (More fine-grained and flexible than previous analyses.)

Intuitive Reductionist Analysis

a. It is obligatory that Bill visits Jill.

b. Bill's obligation to visit Jill is violated.

The violation (fulfilment) expressions appear to predicate of nominalisations of the verbal event.

General question: how to systematically and compositionally create and work with these 'markers' (whether expressed as nominals or otherwise).

Compositionality Example – Sequences of Obligations and Obligations on Sequences

1. It is obligatory that Bill move the left toggle up. After having moved the left toggle up, it is obligatory that Bill move the right toggle left.

There are two obligations on two separate actions; the violations that follow each are distinct and can be used to draw different conclusions.

2. It is obligatory that Bill move the left toggle up, and then move the right toggle left.

This obligation is on the sequence itself (e.g. a protocol); there is one violation, depending on whether or not the protocol is followed in full; there may be various ways to interpret obligations on protocols or sequences. We do not think 1 and 2 are equivalent obligations, nor that we can infer one from the other (in contrast to other theories).

The Action and Deontic Language – Standard Deontic Operator on Action and Sequence

The 'standard' analysis of actions and deontic operators in a Deontic Action Logic. $[\alpha]$ is the action operator (a modal operator from a context where the (weakest) preconditions hold to a context where the postconditions hold); $[\overline{\alpha}]$ is the set-theoretic complement set of actions with respect to α; V is a designated violation marker; ';' is the sequence operator.

a. $\mathbf{OB}\alpha \equiv [\overline{\alpha}](\mathrm{V})$
b. $\mathbf{OB}(\alpha;\beta) \equiv [\overline{\alpha}](\mathrm{V}) \wedge [\alpha]([\overline{\beta}](\mathrm{V}))$

The Action and Deontic Language – [2] Obligation on Action and Sequence

In contrast, we have:

a. $\mathbf{OB}\alpha \equiv [\alpha](\mathsf{F}_{\mathbf{OB}(\alpha)}) \wedge [\overline{\alpha}](\mathsf{V}_{\mathbf{OB}(\alpha)})$

b. $\mathbf{OB}(\alpha;\beta) \equiv [\overline{\alpha}](\mathsf{V}_{\mathbf{OB}(\alpha;\beta)}) \wedge [\alpha]([\beta](\mathsf{F}_{\mathbf{OB}(\alpha;\beta)}) \wedge [\overline{\beta}](\mathsf{V}_{\mathbf{OB}(\alpha;\beta)}))$

The violation (fulfilment) flags are annotated with respect to the complex expression, signalling what has been violated (fulfilled). Prohibitions treated similarly. Issues with permissions in general.

Spell this out as Target:
There is a **deontic specification** which has a deontic operator object **obligation**, an action object **move left toggle up**, an agent object **Bill**, and a deontic flag object **violation**.

The Action and Deontic Language – Pieces

a. DS, the set of deontic specifications.
b. DOL, the set of deontic operator labels, of which "obligation", "permission", and "prohibition" are the only individuals.
c. AcL, the set of action labels, of which "α", "β", and "$\alpha;\beta$" are individuals (among others).
d. AgL, the set of agent labels, of which "Bill" and "Jill" are individuals (among others).
e. DFL, the set of deontic flag labels, of which "violation", "fulfilment", and "null" are the only individuals.

We assume that ds1 and ds2 are *variables* of sort DS.

The Action and Deontic Language – Functions

In addition, suppose the following functions:

a. deonticOpF, a function from objects in DS to objects in DOL.
b. actionF, a function from objects in DS to objects in AcL.
c. agentF, a function from objects in DS to objects in AgL.
d. deonticFlagF, a function from objects in DS to objects in DFL.

The Action and Deontic Language – Deontic Specifications

Target:
There is a **deontic specification** which has a deontic operator object **obligation**, an action object **move left toggle up**, an agent object **Bill**, and a deontic flag object **violation**.

Representation:
$\exists$ x $\in$ DS [deonticOpF(x) = "obligation" $\wedge$ actionF(x) = "α" $\wedge$ agentF(x) = "Bill" $\wedge$ deonticFlagF(x) = "violation"]

The similarity to event-theoretic semantics is *deliberate*.

The Action and Deontic Language - Obligations on Agentive Actions

OB$_{atomic}(\alpha,\iota) =_{def}$

[α](∃ds1 (deonticOpF(ds1) = "obligation" ∧ actionF(ds1) = "α" ∧ agentF(ds1) = "ι" ∧ deonticFlagF(ds1) = "fulfilment")) ∧

[$\overline{\alpha}$](∃ds2 (deonticOpF(ds2) = "obligation" ∧ actionF(ds2) = "α" ∧ agentF(ds2) = "ι" ∧ deonticFlagF(ds2) = "violation"))

for $\alpha \in$ AcL, $\iota \in$ AgL, ds1 and ds2 are variables with values in DS, and ds1 $\neq$ ds2.

"Permission" uses "null" for a deonticFlag.

The Action and Deontic Language - Obligations on Sequences

The language is flexible, allowing us to different notions of obligations on sequences (Interruptable, Collective, and Distributed); the differences are a matter of when a violation (fulfilment) arises and with respect to which action (or combination of actions). They have different purposes. We give the collective interpretation for obligations on *protocols*.

$\mathbf{OB}_{coll}((\alpha;\beta),\iota) =_{def}$

$[\overline{\alpha;\beta}](\exists$ds3 (deonticOpF(ds3) = "obligation" $\wedge$ actionF(ds3) = "$\alpha;\beta$" $\wedge$ agentF(ds3) ="ι" $\wedge$ deonticFlagF(ds3) = "violation")) $\wedge$

$[\alpha;\beta](\exists$ds4 (deonticOpF(ds4) = "obligation" $\wedge$ actionF(ds4) = "$\alpha;\beta$" $\wedge$ agentF(ds4) = "ι" $\wedge$ deonticFlagF(ds4) = "fulfilment"))

The Action and Deontic Language - Obligations on Sequences

Other issues not discussed here:

- The CTD problem (but breaks unwanted inference).
- Interdefinability (between operators), consistency (of sets of deontic specifications), inference (from one expression to another).
- Action negation.

Proposed Analysis in a Process Graph

Given the deontic specifications (aka deontic flags), we need to integrate them into an overall model of how actions, agents, and deontic flags might be animated in a dynamic system.

Agents execute deontically specified actions which give rise to violations or fulfilments with respect to the action and in a context. The context changes, recording what was done, by whom, and the deontic consequences (violations or fulfilments). Deontic specifications can change (introducing new ones, eliminating old ones).

A sample of the Process Graph.

Proposed Analysis in a Process Graph

- Agents (AC), Actions (AC), States-of-Affairs (properties and time index; SOA)
- Valued Action Specifications (VAS) (fine-grained reductions (aka deontic flags) in a Valued Specification List (VSL).
- *Bill is obligated to move the left toggle up* reduced to a VSL comprised of two VASs: *Bill's moving the left toggle up fulfils his obligation to move the left toggle up* and *Bill's moving the left toggle down violates his obligation to move the left toggle up*.
- The VSL is generated by the application of the deontic operator (e.g. **OB**) on the agentive action (e.g. Bill moves the left toggle up).

Proposed Analysis in a Process Graph

- Record what is done, by whom, and the deontic flags in a History (H) in a Historical Record (HR) (A DB of H).
- To represent *contractual breach* and its *remedy* (CTDs), we have Rule Functions (RF) which, given a VAS in the H, updates (introduces or deletes) a VAS in VSL.
- The VSL and RF represents the essence of a *legal* contract.
- Most general component Value Specified Context (**VSC**), which is comprised of the **SOA**, the **VSL**, the **RF**, and the **H**.

Proposed Analysis in a Process Graph - Open Graphic

Simplified illustration of a conceptual process (actions in the program are abstract), assumes two possible actions one of which has to be executed, and has other assumptions about updates.

Can handle complex actions in the abstract language (e.g. sequential) in a *productive* manner. (But will not show this as it requires discussion of the combinators, action negation, etc).

What To Do Next

- How can we relate the language here to natural language syntax and semantics?
- Is an action language needed, or can we further develop event-semantics (which has been done somewhat, but was lacking important components)?
- If our target "market" is legislation or business process modelling, what do we need?
- Spell out the uses and purposes of having an implemented deontic language for risk assessment and compliance.

Thanks for your attention!

Questions

References

Paul McNamara.
Deontic logic.
In Dov M. Gabbay and John Woods, editors, *Handbook of the History of Logic, Volume 7*, pages 197–288. Elsevier, 2006.

Adam Zachary Wyner.
Violations and Fulfillments in the Formal Representation of Contracts.
PhD thesis, Department of Computer Science, King's College London, 2008.