Introduction

Deontic logic

- General goal
  - Design a language for reasoning about norms
    - Greek déon, ‘that which is binding, right’
- Requirements
  - Formal semantics
  - Complete axiomatic characterization
    - Consistency proof: prerequisite for implementation
- Guideline
  - Start with the simplest possible syntax
  - Reserve more complex machinery until the exact limits of the more spartan one are clear
- In this tutorial: no time, no bearers of obligations

Dyadic Deontic Logic

- Introduced by Hansson in 1969 under the label DSDL (Dyadic Standard Deontic Logic)
- Motivation: contrary-to-duty (CTD) obligations
- Full account in Åqvist (2002)

Syntax and Semantics

Layout

- Introduction
- Syntax and semantics of dyadic standard deontic logic
- Meta-theory of dyadic standard deontic logic
**Language**

Syntax of propositional logic

- New building blocks
  - $\Box(B/A) = B$ is obligatory, given $A$
  - $P(B/A) = B$ is permitted, given $A$

A and $B$ are propositional letters

**Semantics**

- Possible worlds (i.e., valuations) are noted $x$, $y$, etc.
- A binary relation $\succeq$ (read "greater than or equal to") is used to rank all the possible worlds $x$, $y$, ..., in terms of betterness.
- Truth-conditions
  - $\Box(B/A)$ true at $x$ iff all the best (according to $\succeq$) $A$-worlds are $B$-worlds
  - Similarly for $P(B/A)$ (but with $\forall$ replaced by $\exists$).

**Example**

- For an unconditional norm, use $\top$ for the condition

- P dual of $\Box$, i.e., $P(B/A) = \neg \Box(\neg B/A)$

**Example**

- Meaning of $\Box A$

**Example**

- Meaning of $\Box A, \Box(B/\neg A)$

- Violation set of $V(x) = \text{set of norms that are violated in } x$

- Put $x \succ y$ iff $V(x) \subset V(y)$
Example

$\begin{align*}
n_1 : \Box A \\
n_2 : \Box(B/\neg A) \\
\end{align*}$

(no not)

SDL-ish binary classification of states into good/bad (green/red) ones too crude

Classes of structures

Constraints on $\succeq$

- Reflexivity: $x \succeq x$
- Transitivity: $x \succeq y$ and $y \succeq z$ implies $x \succeq z$
- Totalness: $x \succeq y$ or $y \succeq x$
- Limit assumption: no infinite sequence of strictly better worlds

Meta-theory

Partial pre-order
Total pre-order
Limit assumption assumed

Meta-theory

Output: set of well-formed formulae (wffs) identified

Language design

alphabet + formation rules

Output: set of well-formed formulae (wffs) identified

Semantics

Logical truth in virtue of logical form

Syntactic consequence: $\vdash$

Axiomatization

Syntactic consequence: $\vdash$

Success criterium

Completeness theorem: $\Gamma \vdash A$ iff $\Gamma \models A$
**Total order case**

- Axiomatization problem
  - Weak completeness result ✓
  - Spohn (1975)
  - Åqvist (1987): system $\mathcal{G}$
  - Strong or full completeness ✓
  - Parent (2008)
- Consistency ✓
- Decidability ✓
- Spohn (1975)

**Partially ordered case**

- Partial pre-order
- Allowing for conflicts between obligations
  - $\diamondsuit A, B$ for $A \succ B$ and $\sim A, \sim B$ for $A \sim B$
  - $\Box(B/A), \Box(\sim B/A)$ both in

**Non-transitive case**

- Call $x$ and $y$ equally good ($x \simeq y$) if $x \succeq y$ and $y \succeq x$

- Argument form
  - If $\succ$ transitive, then $\simeq$ transitive
  - $\simeq$ not transitive
  - So $\succ$ not transitive

- Sorites argument
  - 1000 cups of coffee: $C_1, C_2, C_3, \ldots, C_{999}$
  - $C_1 \simeq C_2 \simeq \ldots \simeq C_{999}$

**Partially ordered case**

- Partial pre-order
- Allowing for conflicts between obligations
- Axiomatization problem
  - Strong & weak completeness: ✓
  - Goble (2003): system DP
  - $\diamondsuit A \rightarrow \neg(\Box(B/A) \land \Box(\sim B/A))$ out (◇: 'possible')
  - Consistency ✓
  - Decidability?

**Non-transitive case**

- Call $x$ and $y$ equally good ($x \simeq y$) if $x \succeq y$ and $y \succeq x$

- Argument form
  - If $\succ$ transitive, then $\simeq$ transitive
  - $\simeq$ not transitive
  - So $\succ$ not transitive

- Modus Tollens
  - If $P$, then $Q$
  - If $\neg Q$, then $\neg P$
  - Therefore, $\neg P$

- Sorites argument
  - 1000 cups of coffee: $C_1, C_2, C_3, \ldots, C_{999}$
  - $C_1 \simeq C_2 \simeq \ldots \simeq C_{999}$
Non-transitive case

Call \( x \) and \( y \) equally good (\( x \simeq y \)) if \( x \succeq y \) and \( y \succeq x \).

Argument form
If \( \succeq \) transitive, then \( \simeq \) transitive
If \( P \), then \( Q \)
\( \simeq \) not transitive
So \( \succeq \) not transitive
Therefore, not-\( P \)

Sorites argument

1000 cups of coffes: \( C_0, C_2, C_3, \ldots, C_{999} \)

\( C_0 \simeq C_2 \simeq C_4 \simeq \ldots \simeq C_{999} \) but \( C_0 \not\simeq C_{999} \)

Non-transitive case

Preliminary result: Parent, to appear: Strong completeness result using an alternative language

Operator: \( QA \) “ideally \( A \)"

\( \Box (B/A) = \Box (QA \rightarrow B) \)

Open problems:
Axiomatize the logic using conditional obligation
Show decidability
On-going work with J. Carmo

Bibliography (1)


Bibliography (2)