Deontic Logic

Legal / Normative Reasoning

Deontic Logic

department of mathematics and computer science
FU Berlin

seminar: legal / normative reasoning

docent: Prof. Christoph Benzmüller

speaker: Thomas Harms

Overview

1. Introduction
    1. SDL – Standard Deontic Logic
    2. Weaknesses
        1. CTD Obligations - Chisholm´s Paradox
        2. Ross´s Paradox
2. Notions of Normative Ideality and Sub-Ideality
    1. Intuitive Idea
    2. SDL Extension – DL
    3. DL Extension – DL*
3. DEON^{+}
    1. ALP & CLP
    2. DEON^{+}
4. Conclusion

1. Introduction Normative Reasoning

- Norms represent desirable behaviour of members of any society
- Normative systems regulate multi-agent systems.
- While preserving agents autonomy, agents might comply or not
- Problems spawned the very active fields of research of normative multi-agent systems [11]
- Norms are encoded in some logic language
    - Components represent notions of obligation, permission and prohibition
    - One way is by means of modal operators in tradition of Deontic Logic

1. Standard Deontic Logic [1]

- Monadic Modal Logic build around the concepts of obligation, permission, option, impermissibility and omission.

- Basic Deontic Axioms are K, D and TAUT
- SDL is often axiomatized as follows:
  - A1. All tautologous wffs of the language (TAUT)
  - A2. $\mathbf{OB}(p \rightarrow q) \rightarrow (\mathbf{OB}p \rightarrow \mathbf{OB}q)$ (**OB**-K)
  - A3. $\mathbf{OB}p \rightarrow {\sim}\mathbf{OB}{\sim}p$ (**OB**-D)
  - R1. If $\vdash p$ and $\vdash p \rightarrow q$ then $\vdash q$ (MP)
  - R2. If $\vdash p$ then $\vdash \mathbf{OB}p$ (**OB**-NEC)

2. Weaknesses of SDL

- Applying Contrary-To-Duty (CTD) norms [10]
- Ross´s Paradox [3]
- Paradox of Kant´s Law
  - Obligation implies ability. Everything you should do, you must be able to do.
  - 1) If you borrow my car, you are obligated to give it back.
  - 2) You crashed my car.
- Free Choice Permission Paradox

2.1. CTD Norms

- Chisholm´s Paradox [8]
- (1) it ought to be that Alice helps her neighbors
- (2) it ought to be that if Alice helps her neighbours, she tells them that she is coming
- (3) if Alice does not help, she ought not to tell them that she is coming
- (4) Alice does not help her neighbours
- Any plausible formalization in SDL turns out to be either inconsistent or not logically independent, which is very undesirable
- Main issue arise from the interpretation of the operator O [12]
    - Semantic intuition of O(A) is that A is true in all normatively ideal worlds

2.1. Ross´s Paradox

- Provable schema in SDL is:

$$\mathbf{OA \rightarrow O(A \vee B)}$$

- **OA**: Bob ought to talk to his wife.
- **O(A ∨ B)**: Bob ought to either talk to his wife or kill his wife.
- In a formalization of a normative system the inference is at least difficult to justify.

Notions of Normative Ideality and Sub-Ideality

- Majority of sentences describing propositions true in all normatively ideal circumstances are not normatively relevant.
- Proposal made by Jones, Pörn [12] requires an 'ought' sentence to describe a proposition:
    - holds in all normatively ideal circumstances
    - fails in all normatively sub-ideal circumstances
    - fails in those circumstances in which not every prescription is observed
- Alice ought to help the neighbours, then this happens in all normatively ideal scenarios, but fails in some sub-ideal scenario.

SDL Extension - DL

- [12] and [3] extend SDL with an operator O` to distinguish ideal cases
- O`A means: A is true in all normatively sub-ideal worlds.
- Formalization of 'ought'-sentences: **Ought(A)** $=_{def}$ **OA ∧ ¬O`A.**
- System DL is obtain by adding O` representing bimodal version of SDL supplemented with the axiom-schema: **(OA ∧ ¬O`A) → A**. [13]
- Frame to interpret DL are structured as: $\mathbf{F} = \langle \mathbf{W}, \mathbf{R_O}, \mathbf{R_{O`}} \rangle$
    - W – domain of possible worlds
    - R_O and $R_{O`}$ are binary relations over W satisfying the properties:
        - for all $w \in W$, there are $v, u \in W$ s.t. wR_Ov and $wR_{O`}u$;
        - for all $w \in W$, $R_O(w) \cap R_{O`}(w) = \emptyset$;
        - for all $w \in W$, either $w \in R_O(w)$ or $w \in R_{O`}(w)$

SDL Extension – DL

- Despite its broader expressive power, DL struggles with some CTD-formalizations.
  - P – Alice helps her neighbours
  - Q – Alice tells her neighbours that she is coming

1. $Ought(P)$
2. $O(P \rightarrow Ought(Q)) \wedge O`(P \neg Ought(Q))$
3. $O(\neg P \rightarrow Ought(\neg Q)) \wedge O`(\neg P \rightarrow Ought(\neg Q))$
4. $\neg P$

- Problem is hidden in the fact, that in DL 1-4 entail both OP and O¬Q, meaning Alice helps her neighbors without telling them that she is coming.

SDL Extension – DL

- Certain other paradoxes of deontic reasoning still hold in DL, such as the version of Ross`s paradox [3]:
  - If you neither return my car nor crash it, while you ought to return it, then you ought to return it or crash it.
  - The schema: $(\neg A \wedge \neg B) \rightarrow (\mathrm{Ought}(A) \rightarrow \mathrm{Ought}(A \vee B))$ is provable in DL.
- [13] proposes a system in order to remedy the latter problem by preventing it to be provable.

DL Extension – DL*

- The operator Ought is replaced with an operator **Ought*(A) =$_{\mathbf{def}}$ OA ∧ O`¬A**.
- *Ought** will be true at a world only if O`¬A is true there.
- The meaning of O` needs to be altered to: A holds in all *normatively awful* worlds.
- DL* denotes the logic resulting from DL by removing axiom (OA ∧ ¬O`A) → A and

  adding the definition of *Ought**.
- The pragmatic oddity of Chisholm`s paradox is still holding in some formalizations.

Abductive Logic Programming (ALP) & Constraint

- Abductive Reasoning – truth value of premises is not bound to truth value of conclusion.
- In an abductive logic program, a distinguished set of predicates, called abducibles, do not have a definition, but their truth value can be assumed.
- A set of formulae, called *Integrity Constraints*, restrict the set of hypotheses in order to avoid unrealistic hypotheses.
- ALP supports hypothetical reasoning and simple, sound implementation of negation by failure.

Deon^{+} - Introduction [2]

- Language where two basic operators (obligation, prohibition) are enriched with quantification over time.
- Syntax is built upon action language, where positive actions are represented by terms, e.g. *answer/2, smoke/1, use/2, drive/2*.
- Terms can contain terms, variables and constants, e.g. *answer(john, me)* stands for the action of answering me, performed by John.
- Negative actions are represented by *not(Action)*, e.g. *not respect(john, speedlimit))*.

Deon^{+} - Obligation [2]

- Obligation are represented as formulas: *obl(A, T)*, where *obl/2* is an abducible predicate, A is any (positive or negative) action and T is a CLP variable possibly (existentially or universally) quantified.
- “It is mandatory, that John answer me”: $\exists T$ *obl(answer(john, me), T)*
- “It is mandatory, John respects the speed limits”: $\forall T$ *obl(respect(john, speed limit), T)*

Deon^{+} - Prohibition [2]

- Similar to obligation, prohibitions are represented as formulas: *forb(A, T)*, where *forb/2* is an abducible predicate, A is any (positive or negative) action and T is a CLP variable possibly explicitly (existentially or universally) quantified.
- "A process p cannot consume all the CPU time": $\exists T$ *forb(use(p, cpu), T)*
- "It is forbidden that John smokes": $\forall T$ *forb(smoke(john), T)*

Deon^{+} - Variables and Constraints [2]

- Language is not limited to the propositional case, as variables can be included possibly explicitly quantified, e.g. "It is forbidden to smoke": $\forall X, \forall T, forb(smoke(X), T)$.
- Adoption of CLP vairibles for representing time adds expressiveness to deontic operators, e.g. covering deadlines by constraints over time.
  - "It is fobidden that John leaves the meeting before 10":

    $\forall T : T < 10\ forb(leave(john,meeting), T)$
  - "It is forbidden, that John leaves the meeting before it ends":

    $\exists T', \forall T : T < T'\ forb(leave(john,meeting), T), end(meeting, T')$

Deon^{+} - Syntax [2]

Definition: A Deon^{+} specification consists of an (abductive) logic programming set of clauses (P), a set of integrity constraints (IC), and a goal (G). A set of meta-level integrity constraints (M) can be added to express or vary the semantics of deontic operators.

- Syntax of deontic literals:

    - *DeonticLiteral ::= [not]DeonticAtom*
    - *DeonticAtom ::= obl(Term, Time)*
    - *DeonticAtom ::= forb(Term, Time)*
    - *Time ::= Variable | Number*

Deon^{+} - Syntax [2]

- Syntax of P and IC:
  - P ::= [Clause]*
  - Clause ::= Atom ← QConjunction.
  - IC ::= [IntegrityConstraint]*
  - QConjunction ::= [ExistentialQ]*[UniversalQ]*Conjunction
  - Conjunction ::= Literal, Conjunction | Literal
  - ExistentialQ ::= ∃V ariable[: Constraints]
  - UniversalQ ::= ∀V ariable[: Constraints]
  - Constraints ::= Constraint | Constraint,Constraints
  - Literal ::= DeonticLiteral | Constraint | DefinedLiteral
  - IntegrityConstraint ::= QConjunction → Disjunction.
  - Disjunction ::= false | QConjunction | QConjunction ∨ Disjunction
  - G ::= ← QConjunction

Deon^{+} - Conditional Obligations and Deadlines [2]

- Integrity constraints can be exploited to represent conditional obligatoriness and the deontic logic of deadlines.
- Conditional obligations can be simply represented by integrity constraints:
  - e.g. B → Obl A is suitable to represent dyadic deontic operator Obl(A|B) [2],[6]
- Deontic logics with deadlines and the operator $O(\rho \leq \delta)$ meaning that action ρ ought to be brought about before another action δ happens.
  - mapped into integrity constraint: $hap(\delta, T_\delta) \rightarrow obl(\rho, T_\rho : T_\rho \leq T_\delta)$

Conclusions

- In order to tackle the usual weaknesses (paradoxes) of SDL we discussed 2 major perspectives by adding expressiveness to deontic operators:
- Distinguishing between normative ideality and sub-ideality in different forms such as DL and DL*.
- Formalization of a language Deon^{+} by exploiting Abductive Logic Programming and Constraint Logic Programming to add precision towards deontic logic operators and expressivity.

Sources

1. https://plato.stanford.edu/entries/logic-deontic/, 12.01.2020
2. Marco Alberti, Marco Gavanelli, Evelina Lamma: "Deon+: Abduction and Constraints for Normative Reasoning”; Chapter out of: A. Artkis, R. Craven, N Kesim, B. Sadighi, K. Stathis: "logic programs, norms and action, Essys in honor of Marek J. Sergot on the occasion of his 60th birthday", Springer 2012, DOI 10.1007 /978-3-642-29414-3_17
3. Tomer Libal, Mateo Pascucci: "Automated Reasoning in Normative Detachment Structures with Ideal Conditions", 10/2018, htp://arxiv.org/pdf/1810.09993
4. Christoph Benzmüller, Ali Farjami, Xavier Parent, "A Faithful Semantic Embedding of the Dyadic Deontic Logic in HOL", 2018, https://christoph-benzmueller.de/papers/R59.pdf
5. Giada Maggenti, Andrea Bracciali, Paolo Mancarella: "Abduction and Legal Reasoning”, 2008

Sources

6. Christoph Benzmüller, Ali Farjami, Xavier Parent: A Dyadic Deontic Logic in HOL, In Deontic Logic and Normative Systems --- 14th International Conference, DEON 2018, Utrecht, The Netherlands, 3-6 July, 2018 (Jan Broersen, Cleo Condoravdi, Shyam Nair, Gabriella Pigozzi, eds.), College Publications, volume 9706, pp. 33-50, 2018
7. J. Carmo and A.J.I. Jones. Completeness and decidability results for a logic of contrary-to-duty conditionals. J. Log. Comput., 23(3):585–626, 2013
8. R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. Analysis, 24:33–36, 1963
9. J. Carmo and A.J.I. Jones. Deontic logic and contrary-to-duties. In D. M. Gabbayand, F. Guenthner, editors, Handbook of Philosophical Logic: Volume 8, pages 265–343. Springer, Netherlands, Dordrecht, 2002
10. Navarro, P.E. and Rodrguez J.L. (2014) `Deontic logic and legal systems'. Cambridge University Press

Sources

11. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational and Mathematical Organization Theory 12, 71–79 (2006)
12. Jones, A. and Pörn, I. (1985), Ideality, sub-ideality and deontic logic. Synthese 65(2), 275-290
13. de Boer, M., Gabbay, D.M., Parent, X. and Slavkovic, M. (2012), `Two dimensional Standard Deontic Logic [including a detailed analysis of the 1985 Jones-Pörn deontic logic system]`. Synthese 187(2), 623-660