A rule-based deontic reasoner

X. Parent
RuleML Webinar 26th January 2018
University of Luxembourg
Department of Computer Science

1 Joint work with L. van der Torre
Talk layout

1. Introduction
2. Benchmark problems
3. Our tool
4. How our solution works (roughly)
5. Conclusion
Background: AI & law

EU Horizon 2020 research and innovation programme–Marie Skodowska-Curie
Grant agreement No 690974.
Introduction

Mining and reasoning

A legal text

Norms (in machine-readable format)

Knowledge base

(Automated) Deontic reasoner

Obligations
Permissions
Legal interpretations
Etc

data

A rule-based deontic reasoner
Introduction

Mining and reasoning

Long term-goal: automated tool support
Norm-compliance checking
Consistency checking
Example

Risk fines of up to €20 million on 25th May 2018 global turnover

How do I Comply with the new EU GDPR?
www.na.com
L. Robaldo

A rule-based deontic reasoner
Deontic logic

- Concerned with obligation, permission and related concepts
- Normative reasoning in law
  - Sergot, McCarthy, Jones, Governatori, Sartor, ...
  - Law as a logical theory

Two research traditions

- Possible worlds semantics (mid 50s)
  - Deontic logic as a branch of modal logic
- “Norm-based” semantics (Hansen, 00s)
  - Roots in Alchourrón and Bulygin’s approach to normative systems

A rule-based deontic reasoner
Deontic logic

- Concerned with obligation, permission and related concepts
- Normative reasoning in law
  - Sergot, McCarthy, Jones, Governatori, Sartor, ...
  - Law as a logical theory

Two research traditions

- Possible worlds semantics (mid 50s)
  - Deontic logic as a branch of modal logic
- “Norm-based” semantics (Hansen, 00s)
  - Roots in Alchourrón and Bulygin’s approach to normative systems
[Parent and van der Torre, 2017]

**Parent, X. and van der Torre, L. W. N. (2017).**
The pragmatic oddity in a norm-based deontic logic.
Introduction

[Parent and van der Torre, 2017]

The pragmatic oddity in a norm-based deontic logic.

Highlights

A “new” logic—in the rule-based tradition (I/O logic)

- Well-defined
- Performs well w.r.t. benchmark problems of deontic logic
  - Contrary-to-duty (CTD) reasoning
  - Conflict
    (Perform well=return the expected answers to queries)
Introduction

[Parent and van der Torre, 2017]

**Parent, X. and van der Torre, L. W. N. (2017).**
The pragmatic oddity in a norm-based deontic logic.

**Highlights**

**Advantage**

- Simple (on the outside)
- User-friendly
  
  Easy to use for non-experts
The pragmatic oddity in a norm-based deontic logic.

**Highlights**

**Novelty 1**
- Consistency checks in the semantics
- Reflected in the proof theory

**Spin-off**
- Handles a recurrent objection against rule-based systems (e.g., Reiter’s default logic): lack of a proof-theory
[Parent and van der Torre, 2017]

**Introduction**

**Parent, X. and van der Torre, L. W. N. (2017).**
The pragmatic oddity in a norm-based deontic logic.

**Highlights**

**Novelty 2**
- A modular treatment of the two categories of benchmarks
  - A unique formalism, not two (separate) formalisms

**Spin-off**
- The reasoner able to handle *both* CTDs and conflicts in the text
Group 1: CTD

The standard CTD structure (Chisholm)

(1) $a$ is obligatory
(2) If not-$a$, then $b$ is obligatory
(3) If $a$, then not-$b$ is obligatory
Introduction

Benchmark problems

Our tool

How our solution works (roughly)

Conclusion

Benchmark problems

Group 1: CTD

The standard CTD structure (Chisholm)

1. $a$ is obligatory
2. If not-$a$, then $b$ is obligatory
3. If $a$, then not-$b$ is obligatory

Primary obligation

CTD obligation

ATD obligation

A rule-based deontic reasoner
Group 1: CTD

The standard CTD structure (Chisholm)

1. $a$ is obligatory
2. If not-$a$, then $b$ is obligatory
3. If $a$, then not-$b$ is obligatory
4. Not-$a$

In the old days: SDL

KD modal logic

Obligatory: true in the ideal worlds

Problem

Formalisation

- Inconsistent

Overall: problem solved. But there are still problems on the periphery.
### Example 1

- Personal data shall be processed lawfully (Art. 5). For example, the data subject must have given consent to the processing of his or her personal data for one or more specific purposes (Art. 6/1.a).
- If the personal data have been processed unlawfully (none of the requirements for a lawful processing applies), the controller has the obligation to erase the personal data in question without delay (Art. 17.d, right to be forgotten).
**Group 2: conflicts (cf. [Goble, 2013])**

**Normative conflict**

The agent ought to do each of several things, but cannot do them all

---

**Goble, L. (2013).**

Prima facie norms, normative conflicts, and dilemmas.

**Normative conflict**

The agent ought to do each of several things, but cannot do them all

Strict, or simple, conflicts: $\Box a, \Box \neg a$

Binary conflicts: $\Box a, \Box b$ but $\neg \lozenge (a \land b)$

N-ary conflicts (general form): $\Box a_1, \ldots \Box a_n$ but $\neg \lozenge (a_1 \land \ldots \land a_n)$

---

**Goble, L. (2013).**

Prima facie norms, normative conflicts, and dilemmas.

Benchmark problems

Group 2: conflicts (cf. [Goble, 2013])

Normative conflict

The agent ought to do each of several things, but cannot do them all

Conflicts across regulations are common-place:

Must the user expressly provide consent to be tracked?

- GDPR: yes
- New EU e-privacy directive: no.

The reasoner must be able to detect/accommodate inconsistencies.

Prima facie norms, normative conflicts, and dilemmas.
Benchmark problems

Group 2: conflicts (cf. [Goble, 2013])

Normative conflict

The agent ought to do each of several things, but cannot do them all

State of the art with the modal logic approach: either the logic is too strong or too weak:

- Too strong: $\Box a, \Box \neg a \vdash \Box b$ (deontic explosion)
- Too weak: $\Box (a \lor b), \Box \neg a \nvdash \Box b$

Also missing: an integration with CTDs.

Prima facie norms, normative conflicts, and dilemmas.
I/O logic (in a nutshell)

One of the success stories of deontic logic

- Devised by Makinson & van der Torre
- Dedicated chapter in the Handbook of Deontic Logic

Conditionals (deontic reading): “If $a$, then $b$ (obligation)”

- Semantics: “operational”
  - Procedures yielding outputs for inputs
- Proof-theory: generalizes existing ones
  - No axiom of identity (““If $a$, then $a$”” )
  - Principle not desirable under a deontic reading

A rule-based deontic reasoner
A parenthesis on LegalRuleML

NOTEWORTHY

List of requirements (from the Oasis LegalRuleML TC)

Support for modelling:

① Different types of rules
  • Prescriptive rules: obligations + permissions
  • Constitutive rules (counts-as conditionals)
    Legal definitions (‘sensitive personal data’)
    Legal interpretation

② Defeasible reasoning (reasoning about exceptions)

I/O logic ticks all the boxes!

→ Well-grounded in the legal domain
A parenthesis on LegalRuleML

**NOTEWORTHY**

**List of requirements (from the Oasis LegalRuleML TC)**

Support for modelling:

① Different types of rules
   - Prescriptive rules: obligations + permissions
   - Constitutive rules (counts-as conditionals)
     Legal definitions (‘sensitive personal data’)
     Legal interpretation

② Defeasible reasoning (reasoning about exceptions)

I/O logic ticks all the boxes!

Well-grounded in the legal domain
Operational semantics

General form

Below: a conditional obligation in the I/O notation

\[(a, x) \mid \{ z \}\]
also called a ‘rule’

\[a\] and \[x\] : two formulae of a base logic

A normative system \(N\) is a set of such pairs.

Main “semantical” construct:

\[x \in O(N, a)\]

\(Given\ input\ a\ (state\ of\ affairs), \ x\ (obligation)\ is\ in\ the\ output\ under\ norms\ N\)

Detachment (modus-ponens) : core mechanism of the semantics
Operational semantics

General form

Below: a conditional obligation in the I/O notation

\[(a, x) | \{z\}\]

also called a ‘rule’

A normative system \(N\) is a set of such pairs.

Main “semantical” construct:

\(x \in O(N, a)\)

Given input \(a\) (state of affairs), \(x\) (obligation) is in the output under norms \(N\)

Detachment (modus-ponens) : core mechanism of the semantics
Operational semantics

General form

Below: a conditional obligation in the I/O notation

\[(a, x) \mid \{z\}\]

also called a ‘rule’

\[a\] and \[x\] : two formulae of a base logic

A normative system \(N\) is a set of such pairs.

Main “semantical” construct:

\[x \in O(N, a)\]

Given input \(a\) (state of affairs), \(x\) (obligation) is in the output under norms \(N\)

Detachment (modus-ponens) : core mechanism of the semantics
**Two-sided architecture**

- **Under the hood**
  - Semantics
    - I/O operations

  ![Diagram of a not user-friendly Garfield](image)

  But needed for, e.g., decidability
  And hidden to the eyes of users

- **On the outside**
  - Proof-theory
    - Derivation tree
      - Nodes are pairs of formulas
      - Inference rules

  ![Diagram of a Garfield with a success criteria](image)

  soundness and completeness results

**Success Criteria**

- **A rule-based deontic reasoner**
A potential misunderstanding

“User-friendly” does not mean trivial.
There is more to I/O logic than just deriving pairs from pairs.

When you see written

\[(a, x)\]
A potential misunderstanding

“User-friendly” does not mean trivial.
There is more to I/O logic than just deriving pairs from pairs.

When you see written

\[(a, x)\]

Under the hood, the reasoner has calculated that

\[x \in O(N, a)\]
Our proposal: a new I/O operation

Semantics

Calculating the output: a 3-step procedure

Is there $B \subseteq Cn(A)$ s.t.

$B$ triggers finitely many obligations in $N$?

Yes

Is $x$ equivalent with the conjunction of their heads?

Yes

Is $B$ consistent with the hypothesis that an arbitrarily chosen obligation (among those triggered) is fulfilled?

Yes

$x$ in the output

No

$x$ not in the output

No

$x$ not in the output

No

$x$ not in the output
Our take on the benchmarks (roughly)

Proof-theory

A modular treatment

Weaken the logic, but not too much

The troublemakers:

\[
\begin{align*}
\text{WO} & \quad \frac{(a, x)}{x \vdash y} \\
& \quad \frac{(a, y)}{(a, y)}
\end{align*}
\]

\[
\begin{align*}
\text{AND} & \quad \frac{(a, x)}{(a, x \land y)} \\
& \quad \frac{(a, y)}{(a, y)}
\end{align*}
\]

It is okay to let WO go. It is not okay to let AND go.
Our take on the benchmarks (roughly)

Proof-theory

A modular treatment

Weaken the logic, but not too much

The troublemakers:

\[
\begin{align*}
\text{WO} & \quad (a, x) \quad x \vdash y \\
& \quad (a, y)
\end{align*}
\]

\[
\begin{align*}
\text{AND} & \quad (a, x) \quad (a, y) \\
& \quad (a, x \land y)
\end{align*}
\]

It is okay to let WO go. It is not okay to let AND go.

Against AND: “pragmatic oddity” (Sergot/Prakken)

\[
\begin{align*}
\text{SI} & \quad (\top, \neg d) \\
& \quad (d, \neg d) \\
\text{AND} & \quad (d, s) \\
& \quad (d, \neg d \land s)
\end{align*}
\]

\text{No!}

d: there is a dog
s: there is warning sign
Our take on the benchmarks (roughly)

**Proof-theory**

**A modular treatment**

Weaken the logic, but not too much

The troublemakers:

\[
\text{WO: } \frac{(a, x)}{x \vdash y} \quad \frac{(a, y)}{(a, y)}
\]

\[
\text{AND: } \frac{(a, x)}{(a, x \land y)} \quad \frac{(a, y)}{(a, y)}
\]

It is okay to let WO go. It is not okay to let AND go.

**In support of AND (Horty)**

Norms come from different sources \(\Rightarrow\) aggregation

- \(m\): military service
- \(c\): civilian

\[
\text{R-AND: } \frac{(\top, m \lor c)}{(\top, \neg m \land c)} \quad \frac{(\top, \neg m)}{(\top, \neg m \land c)}
\]

Yes!
Our take on the benchmarks (roughly)

Proof-theory

A modular treatment

Weaken the logic, but not too much

The troublemakers:

\[
\text{WO} \quad \frac{(a, x)}{x \vdash y} \quad \frac{(a, y)}{(a, y)} \\
\text{AND} \quad \frac{(a, x)}{(a, x \land y)} \quad \frac{(a, y)}{(a, x \land y)}
\]

It is okay to let WO go. It is not okay to let AND go.

A middle way

\[
\text{R-AND} \quad \frac{(a, x)}{(a, x \land y)} \quad \frac{(a, y)}{(a, x \land y)} \quad \frac{a \land x \text{ consistent}}{a \land y \text{ consistent}}
\]
Evaluation

Table 1: Deontic benchmark examples

<table>
<thead>
<tr>
<th>N</th>
<th>A</th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\top, \neg k), (k, k \land g)</td>
<td>k</td>
<td>\neg k, k \land g</td>
<td>\bot</td>
</tr>
<tr>
<td>(\top, \neg c), (k, c)</td>
<td>k</td>
<td>\neg c, c</td>
<td>\bot</td>
</tr>
<tr>
<td>(\top, \neg f'), (a, f')</td>
<td>a</td>
<td>\neg f', f'</td>
<td>\bot</td>
</tr>
<tr>
<td>(\top, \neg f), (f, f \land w), (d, f)</td>
<td>d</td>
<td>\neg f, f</td>
<td>\bot</td>
</tr>
<tr>
<td>(r, c')</td>
<td>r \land s</td>
<td>c', \neg c'</td>
<td>\bot</td>
</tr>
<tr>
<td>(r, c'), (s, \neg c')</td>
<td>r \land s</td>
<td>c', \neg c'</td>
<td>\bot</td>
</tr>
<tr>
<td>(\top, p), (\top, \neg p)</td>
<td>\top</td>
<td>p, \neg p</td>
<td>\bot</td>
</tr>
<tr>
<td>(\top, p)</td>
<td>\neg (p \land h)</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>(\top, p), (\top, h)</td>
<td>\neg (p \land h)</td>
<td>p, h, p \land h</td>
<td>p \land \neg h</td>
</tr>
<tr>
<td>(\top, \neg d), (d, d \land p')</td>
<td>d</td>
<td>\neg d, d \land p', \bot</td>
<td></td>
</tr>
<tr>
<td>(\top, \neg (d \land p'))</td>
<td>\neg (d \land p')</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

k: kill  c: cigarette  p: polite
g: gently  d: dog  h: honest
f: fence  r: rain  a: asparagus
w: white  s: sun  c': close
f': finger  p': poodle

Good news: reasoner returns the expected answers.
I have described a deontic reasoner currently under development.

- Well-defined:
  - Semantics and proof-theory
  - Completeness result linking the two
- Performs well on the two categories of benchmark problems of deontic logic

Spin-off: allows us to address a recurrent objection against rule-based systems

- Lack of proof-theory
Future work

A big step forward, but only a first step.

Extensions

- Time, exceptions and other natural language constructs
- Permission and constitutive rules

Automation

- On-going work, with Benzmüller
  - Embedding into Higher-Order Logic (HOL)
  - Theorem-prover Isabelle/HOL for automation
Thank you!
Non-monotonic logics & CTDs

Flavored by, e.g., McCarthy (early 90’s)

**Bottom line**

\[
\text{SI} \quad \frac{(\top, \neg d)}{(d, \neg d)}
\]

Dashed line: the inference is blocked.

Sergot’s view: no good for norm-compliance checking
Non-monotonic logics & CTDs

Flavored by, e.g., McCarthy (early 90’s)

**Bottom line**

\[
\text{SI } \frac{(\top, \neg d)}{(d, \neg d)} \quad \text{Dashed line: the inference is blocked.}
\]

“The non-monotonic properties of a logic program using negation-by-failure make a consistent representation [of CTDs] possible. However, the program will have certain counter-intuitive properties. For instance, violated obligations simply vanish. Nothing more can be inferred about them, as the condition for something being obligatory no longer applies. One might argue that in actual life violated obligations do not vanish.” (Herrestad)