What is Norm Change?

Guido Boella & Gabriella Pigozzi & Leon van der Torre
Approaches to norm change

• **1981**: Conference “Informatica, Logica e Diritto” (Florence)

SESSION 2

Abrogation of Rules and Syntactic Decidability in Legal Orders

*Coordinator: Antonio A. MARTINO*

*Professor of Political Science at the University of Pisa*

The abrogation of rules creates special problems in determining which is the ‘legal system in force’, as in the case of abrogation of the consequences of explicit rules and not of the rules themselves.

One way of solving these problems consists in the reconstruction of a model of ‘legal order’ based on the concept of hierarchy.
§1. **Background.** The simplest and best known form of theory change is *expansion*, where a new proposition (axiom), hopefully consistent with a given theory $A$, is set-theoretically added to $A$, and this expanded set is then closed under logical consequence. There are, however, other kinds of theory change, the logic of which is less well understood. One form is theory *contraction*, where a proposition $x$, which was earlier in a theory $A$, is rejected. When $A$ is a code of norms, this process is known among legal theorists as the *derogation* of $x$ from $A$. The central problem is to determine which propositions should be rejected along with $x$ so that the contracted theory will be closed under logical consequence. Another kind of change is *revision*, where a proposition $x$, inconsistent with a given theory $A$, is added to $A$ under the requirement that the revised theory be consistent and closed under logical consequence. In normative contexts this kind of change is also known as *amendment*. 
• 2001: best paper award to Juliano Maranhao “Refinement - A tool to deal with inconsistencies”.

• Inspired by selective revision of Ferme’ and Hansson (1999), where only part of the input information is accepted.

• The agent refines his knowledge by accepting the new information under certain conditions:

$$K \sqsubseteq \alpha$$ is the partial meet revision $$K^*(\beta \rightarrow \alpha)$$

• Refinement provides a tool to avoid contradictions in normative systems: e.g. cases where judges face new conditions which were not mentioned in the legal statute but turn out to be relevant in the application of legal rules to practical cases.
2009: Dynamic Context Logic

- Approach inspired by the dynamic logic paradigm. It looks at norm change as a form of model update.
- Contexts can be seen as sets of models of given theories.
- Modal logic of context: $[X] \alpha = \text{‘in the context of X, it is the case that } \alpha \text{’}$.
- Context expansion ($X+\alpha$) is linked to norms promulgation and context contraction ($X-\alpha$) to derogation of norms.
- Norms are logical statements like ‘a certain fact $\alpha$ implies a violation’ ($V$).
- Counts-as: $\varphi \Rightarrow_X \psi \overset{\text{def}}{=} [X](\varphi \rightarrow \psi)$
- Obligations: $\Box_X \varphi \overset{\text{def}}{=} \neg \varphi \Rightarrow_X V$
2009: Dynamic Context Logic (cont’d)

• Promulgation and derogation of classificatory rules:

\[+(\varphi \Rightarrow_x \psi) \overset{def}{=} X+(\varphi \rightarrow \psi)\]
\[-(\varphi \Rightarrow_x \psi) \overset{def}{=} X-(\varphi \rightarrow \psi)\]

• Promulgation and derogation of obligations:

\[+O_x \psi \overset{def}{=} X+(\neg \psi \rightarrow V)\]
\[-O_x \psi \overset{def}{=} X-(\neg \psi \rightarrow V)\]

• Some theorems in DCxt:

\[-[X]\psi \rightarrow (\varphi \leftrightarrow [X-\psi]\varphi)\] \textit{minimality} (cfr AGM vacuity: \(\text{If } x \notin K \text{ then } K = K - x\))
\[\alpha \rightarrow [X-\psi][X+\psi] \alpha \text{ for } \alpha \in \mathcal{L}_{Prop}\] \textit{recovery} (cfr AGM: \(\text{If } x \in K \text{ then } K \subseteq (K - x) + x\))
2009: Changing Legal Systems: Legal Abrogations and Annulments in Defeasible Logic

• Syntactic approach, where norm change is an operation performed on the rules contained in the code.
• Inspired by legal practice: annulments and abrogations.
• Annulment makes a norm invalid and removes it from the code. The effect applies *ex tunc*: all effects (past and future) are cancelled.
• Abrogation is somehow removed. The effect applies *ex nunc*: some or all their effects propagate, if obtained before the modification: it cannot operate retroactively.
• “If $n_1$ is abrogated in 2007, its effects are no longer obtained after then. But, if a case should be decided in 2008 but the facts of the case are dated 2006, $n_1$, if applicable, will anyway produce its effects because the facts held in 2006, when $n_1$ was still *in force*. Accordingly, $n_1$ is still in the legal system, even though is no longer in force after 2007”.
2009: Changing Legal Systems (Cont’d)

• First, they try to capture annulments and abrogations with theory revision in DL without temporal reasoning. The result is negative: retroactivity cannot be captured.

• Then, they use a temporal extension of DL.

• Norms have different temporal dimensions: the time of validity of a norm (when the norm enters the normative system) and the time of effectiveness (when the norm can produce legal effects). This is why they need multiple versions of a normative system.
Some open issues

- **Normchange ’07 (Luxembourg):** Most models focus on dynamics of obligations and permissions. Models that represent changes in underlying norms are needed: distinguish norms from obligations and permissions.

- **“Issues in designing logical models for norm change” (2009):** How do norms depend on abilities? Abilities change and so should norms do. Useless to specify norms about behaviors that cannot occur. On the other hand, new abilities may require adaptation of norms.

- **Norms and preferences:** Norms may oppose agents’ preferences (but may be enforced). This may change individual preferences. But also: a norm can be chosen to promote a socially optimal outcome.
Some open issues (Cont’d)

• How is change best represented?

  1. In AGM, the postulates are formulated in a meta-language. Another option: formulate policies for change in the object language (DEL). Which option to choose?

  2. Discrepancy between formalisms for norms (I/O logics) and formalisms which allow a more concrete representations of obligations and permissions (temporal deontic logics, STIT logics).

  3. If norms are represented as rules, then change can also be represented as non-monotonic inference: changing a normative systems would amount to adding new rules.
How to Formally Characterize Norm Change?

• There are various models of norm change
  – How to decide on a common framework?

1. Norms must be distinguished from obligations
  – Static: norms identified (confused) with obligations
  – Dynamic: we need to be more precise

2. Use framework for norm & obligation change
  – Common model for both at same time
  – Norm change like BPvdT:AAMAS09 is projection

Dynamics Forces Us To Be Precise

•

Fig. 1. Labeling of the temporal structure using the normative system \( NS = (CN, RN) \) with \( CN = \{(sign_{jp}, sign_{jp}, contr), (sign_{jp}, owe_{jp}, contr)\} \) and \( RN = \{(owe_{jp}, pay_{jp}, d), (pay_{jp}, receipt_{pj}, e)\} \).
Fig. 1. Labeling of the temporal structure using the normative system \( NS = (CN, RN) \) with \( CN = \{ (esign_{jp}, sign_{jp}, contr), (sign_{jp}, owe_{jp}, contr) \} \) and \( RN = \{ (owe_{jp}, pay_{jp}, d), (pay_{jp}, receipt_{pj}, e) \} \).
IOL and Streams

- Input/output logic for streams
- Output/input logic for channels
- Input/output networks: active / flow

- Example: knowledge based obligation

\[ A_1, A_2, A_3, \ldots \rightarrow K \rightarrow O \rightarrow X_1, X_2, X_3, \ldots \]

IOL and Streams

• Input/output logic for streams
• Output/input logic for channels
• Input/output networks: active / flow

• Example: knowledge based obligation

Summary

• There are various models of norm change
  – How to decide on a common framework?

1. Norms must be distinguished from obligations
   – Static: norms identified (confused) with obligations
   – Dynamic: we need to be more precise

2. Use framework for norm & obligation change
   – Common model for both at same time
   – Norm change like BPvdT:AAMAS09 is projection

• NMR 2010 session on preference and norms (with Frederic Koriche), deadline January 29
  – as shown by work of Guido and Nino, nonmonotonic logic and norm change are related

• ESSLLI 2010 course on deontic logic (with Jan Broersen)
Norms and Obligations

<table>
<thead>
<tr>
<th>Norms (&amp; Imperatives)</th>
<th>Obligations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule</td>
<td>Application</td>
</tr>
<tr>
<td>Law</td>
<td>Case</td>
</tr>
</tbody>
</table>

\[(a, x) \in N \quad x \in out(N, a)\]
Norm Change

\[ N = \{ (a_1, x_1), (a_2, x_2), \ldots, (a_n, x_n) \} \]

- Inspired by AGM theory change:
  - Expansion: \( N' = N \oplus (a, x) \)
  - Contraction: \( N' = N \oslash (a, x) \)
  - Revision: \( N' = N \otimes (a, x) \) (one norm at a time)

- Base change: causal rejection principle (LP)
  - Our challenge: stay close to AGM, “more semantic”
Law2Case Bridge Principle

If $(a,x) \in N$ then $x \in out(N,a)$
Strong Case2Law Bridge Principle

If $x \in \text{out}(N,a)$ then $
\forall Y : \text{out}(N,Y) = \text{out}(N \cup (a,x),Y)$

$x!$
Formalization Bridge Principles

\[(a, x) \in \text{out}(N) \iff x \in \text{out}(N, a)\]

- Reflexivity (Law2Case principle)
  \[N \subseteq \text{out}(N) \quad (a, x) \in N \implies x \in \text{out}(N, a)\]

- Idempotence (strong Case2Law principle)
  \[\text{out}(N) = \text{out}(\text{out}(N))\]
Tarskian Consequence

\[(a, x) \in \text{out}(N) \iff x \in \text{out}(N, a)\]

- Reflexivity (Law2Case principle)
  \[N \subseteq \text{out}(N) \quad (a, x) \in N \Rightarrow x \in \text{out}(N, a)\]

- Monotony
  \[\text{out}(N_1) \subseteq \text{out}(N_1 \cup N_2)\]

- Idempotence (strong Case2Law principle)
  \[\text{out}(N) = \text{out}(\text{out}(N))\]

- \(\text{out}(N)\) are “implicit” rules in normative system
Example: out₁ = simple-minded output
1. (a,x): If input implies a, then output implies x
2. Each out₁(N,In) is closed under “Cn”

Numerous IO logics (seven studied in JPL00)
• Not only new logics, but a new framework

Revising a Sequence

<table>
<thead>
<tr>
<th></th>
<th>In</th>
<th>T</th>
<th>a</th>
<th>¬a</th>
<th>b</th>
<th>a∧b</th>
<th>a∧¬b</th>
<th>a∧b∧c</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>out₁(N,In)</strong></td>
<td>T</td>
<td>x∧y</td>
<td>T</td>
<td>T</td>
<td>x∧y</td>
<td>x∧y</td>
<td>x∧y</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>In</th>
<th>T</th>
<th>a</th>
<th>¬a</th>
<th>b</th>
<th>a∧b</th>
<th>a∧¬b</th>
<th>a∧b∧c</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>out₁(N,In)</strong></td>
<td>T</td>
<td>x∧y</td>
<td>T</td>
<td>z</td>
<td>x∧y∧z</td>
<td>x∧y</td>
<td>x∧y∧z</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>In</th>
<th>T</th>
<th>a</th>
<th>¬a</th>
<th>b</th>
<th>a∧b</th>
<th>a∧¬b</th>
<th>a∧b∧c</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>out₁(N,In)</strong></td>
<td>T</td>
<td>x∧y</td>
<td>T</td>
<td>T</td>
<td>x∧y</td>
<td>x∧y</td>
<td>x∧y</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>In</th>
<th>T</th>
<th>a</th>
<th>¬a</th>
<th>b</th>
<th>a∧b</th>
<th>a∧¬b</th>
<th>a∧b∧c</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>out₁(N,In)</strong></td>
<td>T</td>
<td>x∧y</td>
<td>T</td>
<td>T</td>
<td>x∧y</td>
<td>x∧y</td>
<td>x∧y</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>In</th>
<th>T</th>
<th>a</th>
<th>¬a</th>
<th>b</th>
<th>a∧b</th>
<th>a∧¬b</th>
<th>a∧b∧c</th>
<th>...</th>
</tr>
</thead>
</table>
Revising a Sequence

\( (a, x \land y) \)

\[ \oplus (b, z) \]

\( (a, x \land y), (b, z) \)

\( (a, x), (a, y) \)

\[ \oplus (b, z) \]

\( (a, x \land y), (b, z) \)
The design of complex multi-agent systems is increasingly having to confront the possibility that agents may not behave as they are supposed to. In addition to analysing the properties that hold if protocols are followed correctly, it is also necessary to predict, test, and verify the properties that would hold if these protocols were to be violated. We illustrate how the formal machinery of deontic interpreted systems can be applied to the analysis of such problems by considering three variations of the bit transmission problem. The first, an example in which an agent may fail to do something it is supposed to do, shows how we deal with violations of protocols and specifications generally. The second, an example in which an agent may do something it is not supposed to do, shows how it...
Static Framework?

- AB: “Defeasible Rule update is representable in a static framework of a prioritized set of rules.”

- Causal rejection principle (LP)
  - Syntactic flavor
  - AB: Identity (Dominance), Trans, And
    - no RW: relation with AGM?
    - no OR: horn clauses (e.g. TM)

- Permissions, priorities, etc
  - IOL: distinguish rules and permissions, priorities etc
  - Some open problems, with e.g. Horn clauses
Ten Challenges for NorMAS

1. Tools for agents supporting communities in their task of recognizing, creating, and communicating norms to agents.
2. Tools for agents to simplify normative systems, recognize when norms have become redundant, and to remove norms.
3. Tools for agents to enforce norms. In a distributed approach, roles should be defined for agents in charge of monitoring and sanctioning.
4. Tools for agents to preserve their autonomy.
5. Tools for agents to construct organizations.
6. Tools for agents to create intermediate concepts and normative ontology, for example to decide about normative gaps.
7. Tools for agents to decide about norm conflicts.
8. Tools for agents to voluntarily give up some norm autonomy by allowing automated norm processing in agent acting and decision making.

Ten Guidelines

1. Motivate which definition of normative multiagent system is used.
2. Make explicit why norms are a kind of (soft) constraints deserving special analysis.
3. Explain why and how norms can be changed at runtime.
4. Discuss the use and role of norms as a mechanism in a game-theoretic setting.
5. Clarify the role of norms in the multiagent system.
6. Relate the notion of “norm” to the legal, social, or moral literature.
7. Use norms not only to distinguish right from wrong, but also to resolve dilemmas, and use norms not only describe violations, but in general to coordinate, organize, guide, regulate or control interaction among agents.
8. Distinguish norms from obligations, prohibitions and permissions.
9. Use the deontic paradoxes only to illustrate the normative multiagent system.
10. Consider regulative norms in relation to other kinds of norms and concepts.

Table 6. Ten guidelines for the development of normative multiagent systems