Visualizing Normative Systems:
an Abstract Approach

Silvano Colombo Tosatto, Guido Boella,
Leendert van der Torre, and Serena Villata

[1] SnT Luxembourg
`silvano.colombotosatto@uni.lu`
[2] University of Torino
`guido@di.unito.it`
[3] University of Luxembourg
`leon.vandertorre@uni.lu`
[4] INRIA Sophia Antipolis
`serena.villata@inria.fr`

Abstract. Abstract normative systems allow to reason with norms even when their content is not detailed. In this paper, we propose a our preliminary results to visualize abstract normative systems, in such a way that we are able to reason with institutional facts, obligations and permissions. Moreover, we detect meaningful patterns emerging from the proposed visualization, and we show how these patterns can be used to define commonly used reusable solutions.

Keywords: Abstract normative systems, graph patterns

1 Introduction

Successful reasoning formalisms in Artificial Intelligence such as Bayesian networks, causal networks, belief revision, dependence networks, CP-nets, Dung's abstract argumentation systems, come with intuitive and simple visualizations. Traditionally deontic logic has been associated with preference orders [5], which have an intuitive visualization too. However, it is less clear how to extend this visualization of pre-orders to other aspects of normative reasoning. We therefore raise the following question:

Research question. How to visualize normative reasoning?

In general, we see two approaches to this question, depending on the audience for which the visualization is developed. On the one hand, we may aim to illustrate a derivation in all its details, and on the other hand, we may look for an abstract approach that visualizes the rough structure of normative reasoning, hiding the more detailed structure. Such an abstract approach may be used also to summarize a more complex derivation. In this paper we follow the latter approach. We thus aim for a visualization that can be understood by non-experts in normative reasoning.

The success criterion of our research challenge is that it covers a wide range of topics discussed in the first volume of the handbook of deontic logic [4], namely:

Conditionals and rules, we do not restrict ourselves to monadic approaches, but also capture dyadic ones.

Contrary to duties, well known from deontic paradoxes [1, 3], must be represented without deontic explosion, that is, without derivation of all possible obligations.

Dilemmas, if two norms give contradictory advice, then we must to be able to prevent explosion.

Permissive norms must be represented explicitly and used to generate permissions.

Constitutive norms, such as count-as conditionals, must be represented explicitly, and used to generate institutional facts.

An intuitive and simple visualization for abstract normative systems is important to make them adopted in real applications. The idea shares the motivation with Dung's argumentation networks for non-monotonic reasoning [2], with visual languages as UML for object-oriented software engineering[5] [12], and TROPOS-like visual representation of early and late requirements[6] [9, 10], etc. Assuming the standard visualizations of graphs, still many questions remain, such as:

1. How to visualize the element and its anti-element? How to visualize the context of the norms?
2. How to visualize the derived obligations, permissions and institutional facts?
3. How to visualize patterns like violations, contrary to duties, and dilemmas?
4. More generally, how can we visualize patterns as commonly used solutions which can be reused, similarly to what is presented in abstract argumentation [15]?

Though we promote the use of visual reasoning formalisms for normative reasoning, we insist that the formalism should have a clear and unambiguous semantics. Moreover, we believe that a visual formalism is best accompanied by a logical one, and we therefore refer to visualization of normative reasoning rather than a visual reasoning formalism. In this paper we use a notation that is suited both for printed documents and hand-written notes, however this choice presents some limitations such as the difficulty to represent multiple contexts and outputs, which for instance prevents reasoning by cases.

The rest of this paper is structured as follows. In Section 2, we provide the basic insights on abstract normative systems [14]. Section 3 introduces the visual language we propose, and Section 4 describes the formalization of an example using the proposed visual language. In Section 5, we present the patterns that may emerge from the modeling of normative systems using our visual language and we describe how to use them for extracting the basic normative components of an abstract normative system. Section 6 concludes the paper.

---

[5] http://www.uml.org/

[6] http://www.troposproject.org/node/120

2   Abstract Normative Systems: fundamentals

We [14] recently introduced abstract normative systems as a general framework to compare logics for normative reasoning like Input/Output logic [7] and CIS (Condition Implication Structures) [6]. The advantage of reasoning at the abstract level is that we do not need to detail the content of the norms. We now provide the basic insights of abstract normative systems, for more details see [14].

Definition 1 (Universe L). *Given a* finite *set of atomic elements E, the universe L is $E \cup \{\sim e \mid e \in E\} \cup \{\top\}$. For $e \in E$, let $\overline{a} = \sim e$ iff $a = e$, $\overline{a} = e$ iff $a = \sim e$, and undefined iff $a = \top$.*

As presented in Definition 1, each element in the universe L comes with its "anti-element", which is used to represent violations as the presence of the element in the input and the presence of its anti-element in the output. Moreover, each context contains an element $\top$. An abstract normative system is a directed graph, where the context is the set of nodes in the graph containing $\top$.

Definition 2 (ANS $\langle L, N \rangle$). *An abstract normative system ANS is a pair $\langle L, N \rangle$ with $N \subseteq L \times L$ a set of conditional norms, and a context $A \subseteq L$ is a subset of the universe such that $\top \in A$.*

Abstract normative systems generate obligations sets reflecting the obligatory elements of the universe. In particular, we highlight the following eight operations from input/output logic [7].

Definition 3 (Deontic operation $\bigcirc$). *A deontic operation $\bigcirc$ is a function from an abstract normative system $\langle L, N \rangle$ and a context A to a subset of the universe $\bigcirc(\langle L, N \rangle, A) \subseteq L$. Since L is always clear from context, we write $\bigcirc(N, A)$ for $\bigcirc(\langle L, N \rangle, A)$.*

Definition 4 (Eight deontic operations). *A context $A \subseteq L$ is complete if for all $e \in E$, it contains either e or $\overline{e}$ (or both).*

$$\bigcirc_1(N, A) = N(A) = \{x \mid (a, x) \in N \text{ for some } a \in A\}$$
$$\bigcirc_2(N, A) = \cap\{N(V) \mid A \subseteq V, V \text{ complete}\}$$
$$\bigcirc_3(N, A) = \cap\{N(B) \mid A \subseteq B \supseteq N(B)\}$$
$$\bigcirc_4(N, A) = \cap\{N(V) \mid A \subseteq V \supseteq N(V), V \text{ complete}\}$$
$$\bigcirc_i^+(N, A) = \bigcirc_i(N \cup \{(a, a) \mid a \in L\}, A)$$

Two kinds of permissions are studied, namely negative and static positive permissions, where the former are defined as a deontic operation and the latter is detailed in Definition 5.

Definition 5 (ANS^P). *An abstract normative system with permissive norms ANS^P is a triple $\langle L, N, P \rangle$ with $N, P \subseteq L \times L$ two sets of conditional norms. The positive permission operation P^0 is a function from an $ANS^P = \langle L, N, P \rangle$ and a context A to a subset of the universe $P^0(\langle L, N, P \rangle, A) \subseteq L$.*

3 Visualization

In this section we describe our visual language to represent and reason about norms. First, we introduce the components used by the visual language, including the elements and the different types of norms. Second, we define how the components interact and how permissions and obligations are derived using the visual language.

3.1 Visual components

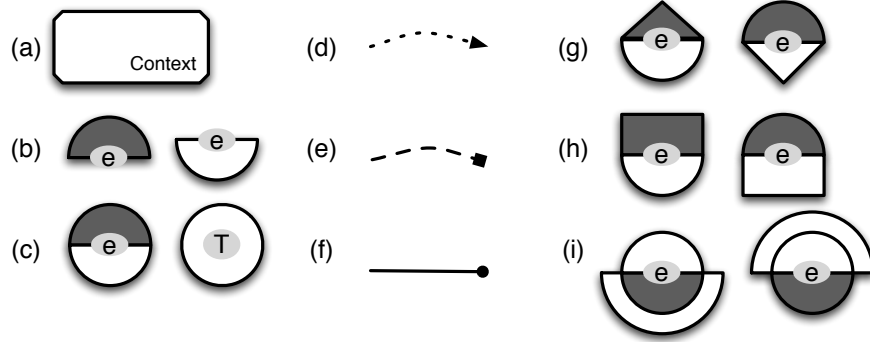In this section we describe the components of the visual language.

Fig. 1. Legend.

The *legend* in Figure 1 introduces the components used in our language. In the picture we can observe the context *(a)*, positive and negative elements *(b)*, entity and $\top$ element *(c)*, permissive norm *(d)*, regulative norm *(e)*, constitutive norm *(f)*, permissions *(g)*, obligations *(h)* and institutional concepts *(i)*.

In the remainder part of this section we provide details for the components of the language.

Entity The basic component of our visual language is the "entity". An entity represents pairs of elements from the *universe*, except for the $\top$ entity which is not a pair but a single element. Every pair constituting an entity is composed by an element e and its negation $\sim e$, the anti-element. An element and its anti-element represent a propositions and its negation. In the paper we also use the following concept of *complement*: considering an element or anti-element φ, the complement $\overline{\varphi}$ is the anti-element if φ is the element and vice versa.

Definition 6 (Entity). *Given a pair of elements e and $\overline{e}$ such that $e, \overline{e} \in E$. We define ⓔ as the entity which contains the element e and the anti-element $\overline{e}$ and Ⓣ as the entity which contains $\top$.*

Context While reasoning about norms, it is also important to consider in which situation the norms are applied. We consider the situation where the norms are applied as the *context*. We can consider a context as a photography capturing the time window that captures the current state of the world. Assuming a dynamic world changing over time, we can figure that the context varies its shape to accommodate the elements that describe the present state as the world evolves. A context encloses different elements at different points in time but it includes always $\boxed{\top}$.

Definition 7 (Context). *Given L as the set representing the universe and a context A. $A = L' \cup \boxed{\top} | L' \subseteq L$.*
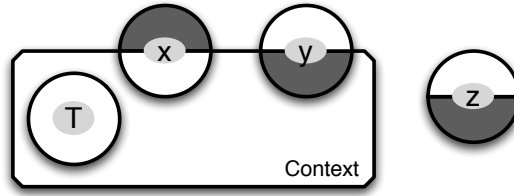

Fig. 2. Elements, Brute Facts and Context.

In Figure 2 we illustrate the entities $\textcircled{x}, \textcircled{y}, \textcircled{z}$ and $\boxed{\top}$ along with a context. The context is represented as a box with smoothed corners that includes the elements which describe a given state of the world. Entities are represented by circles and are labeled according to the element that they represent. Except for $\boxed{\top}$, each other circle representing an entity is divided in two sides. Considering for instance the element $\textcircled{x}$, we consider the light side of the circle as the element x and the dark side as its anti-element $\sim x$. The $\boxed{\top}$ entity is always enclosed within the context, being always part of the elements that describe the current state of the world. In the picture also elements x and $\sim y$ are part of the context and are represented enclosed within the context's boundaries.

Norms Having described the components necessary to visualize the elements involved in a normative system we define now the different types of norms and how they are represented in our visual language.

Following Makinson and van der Torre [7], we represent the norms as conditional rules. Each rule is composed by two distinct components: an antecedent and a consequent. We represent conditional rules as pairs, considering an arbitrary rule r, it is represented as follows: $r = (x, y)$ where x refers to its antecedent and y to its consequent. A conditional rule is considered as activated if its antecedent is present in the context in which the rule is considered. The consequent

describes the consequence of the rule being activated, in our case we reason about norms, thus activating rules produce normative consequences.

We distinguish three different types of norms: regulative, permissive and constitutive norms. In our language we represent norms as arrows originating from the element representing the antecedent and pointing to the consequent. To distinguish the three kinds of norms, the arrows used differ according to the type of norm represented, both on the shape of the head of the arrow and the line used.

An arrow representing a norm can either originate from the element or anti-element side of an entity, depending on which element triggers the rule. The same is true for where the arrow points, with the difference that the pointed element refers to the normative concept produced by the norm. Considering $\top$, an arrow can only originate from it but not point at, meaning that $\top$ can be only considered as antecedent but not consequent. Another difference is that $\top$ does not share the dual nature of the other entities, because it is only composed by an element without its anti-element.

Regulative norms define in the normative system what is obligatory. In our language we do not explicitly represent prohibitions, however we can represent them as complementary obligations, this means that if an element e needs to be forbidden, we introduce the obligation of its complement, in this case $\bar{e}$.

In our visual language we represent rules as arrows between elements, the point of origin of an arrow represents the antecedent and the element pointed by it the consequent. Except for $\top$ is different if an arrow originates or points to the dark or bright side of an element. For instance if an arrow originates from the bright side of $\copyright$, then the antecedent of that rule is e, conversely if the arrow originates from the dark side, then the antecedent is $\sim e$. The same applies to which side of an element an arrow points and the consequent of the rule.

An arrow representing a regulative norm is represented as a dashed line with a squared head to distinguish it from the other types of norms.

In Figure 3 we can see two instances of regulative norms, considering illustration (**a**) we can see a regulative norm the antecedent of which $\sim x$ belongs to the context and its consequent is $\sim y$. Illustration (**b**) of Figure 3 represents a rule from the the element x to the element y. Differently from the previous illustration, in this case the antecedent of the rule is not part of the context.
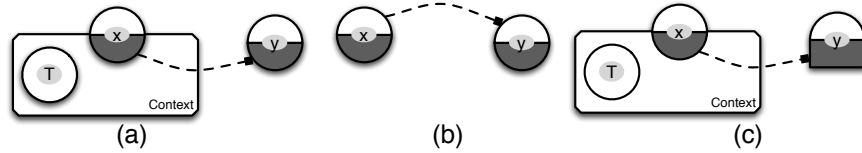

Fig. 3. Regulative norms and obligation.

Regulative norms produce obligations when activated, which are represented in our language, the side of the element which becomes obligatory is changed from the original round shape to a box as shown in Figure 3 illustration (**c**) where $\sim y$ is an obligation. In this way obligations can be easily spotted in the visual representation of the normative system.

The second type of norm we describe is the permissive norm. A permissive norm defines in a normative system what is explicitly permitted in the situation in which the system is considered.

Arrows representing permissive norms are drawn as dotted lines with a triangular head and are used to produce positive permissions in the normative system.

We differentiate two types of permissions of the three described by Makinson [8]: positive and negative permissions. On the one hand we consider something as negatively permitted if there is no obligation of its complement, we do not explicitly visualize negative permissions in our language but we can assume that something is a negative permission if the complementary side of the element is not an obligation.

Positive permissions on the other hand refer to the elements which are explicitly stated as permitted by permissive rules.

In Figure 4 are represented two instances of permissive norms. In illustration (**a**) we can see a permissive norm departing from the element x, belonging to the context, and pointing to $\sim y$. Differently illustration (**b**) shows a permissive rule the antecedent of which, $\sim x$, does not belong to the context.
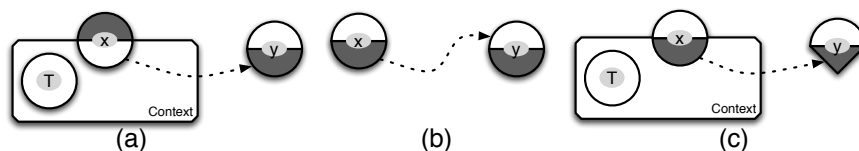

Fig. 4. Permissive norms and positive permission.

Positive permissions are represented in our visual language by changing the shape of the pointed side of an element to a triangle. A representation of a visual positive permission is shown in Figure 4 illustration (**c**) where $\sim y$ is a positive permission.

To justify our choice how to represent obligations, permissions, permissive and regulative norms, we consider modal logic operators for necessity and possibility ($\square$ and $\lozenge$). These two operators can be respectively related to obligations and permissions. For this reason we decided that an intuitive way to picture the arrows, obligations and permission was to associate the $\square$ to the arrows representing regulative norms and obligations, and the $\lozenge$ (we actually use triangles which resemble a split $\lozenge$) for permissive norms and positive permissions.

The ⊤ entity can be used as an antecedent for permissive and regulative norms. For instance if we consider a regulative norm described by the following rule: $(\top, x)$, this rule is activated if $\top$ belongs to the context or is an institutional fact. However according to Definition 7 the entity ⊤ is always part of the context, this means that if a norm (either regulative or permissive) has the element $\top$ as the antecedent then it is always activated.
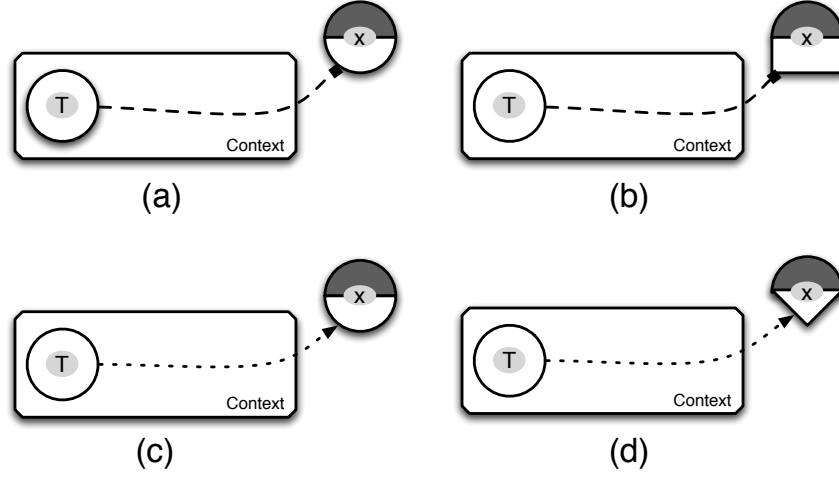

Fig. 5. Regulative and Permissive norms with $\top$ as antecedent.

In Figure 5 we show some instances of norms which are always activated. In illustration **(a)** we visualize a regulative norm with ⊤ as antecedent and x as consequent. Being always activated, the considered norm always produces its obligation as shown in illustration **(b)**. Figure 5 illustrations **(c)** and **(d)** visualize a similar case for a permissive norm which always produces the positive permission associated.

The third type of norms are constitutive norms. Constitutive norms are also known as "counts-as conditionals" rules [13] and are used to derive *institutional facts*. Institutional facts are contrasted by brute facts, which represent the elements describing the state of the world. Differently institutional concepts represent abstract concepts which have meaning in normative reasoning. For example we can consider the action "pickpocketing a wallet" as a brute fact clearly observable in the world. We can then take into consideration constitutive norms, for example "pickpocketing a wallet counts-as a theft" and "a theft counts-as a crime". The constitutive norms introduce two institutional facts: theft and crime which are abstract concept valid if considered within a normative context.

The arrows representing constitutive norms are drawn as straight continuous arrows with a circle head.

In Figure 6 we show two examples of constitutive norms. The first illustration **(a)** shows a constitutive norm having as antecedent the element $\sim x$ which is also a brute fact and as consequent $\sim y$. The second illustration **(b)** shows a constitutive norm the antecedent of which is not a brute fact.
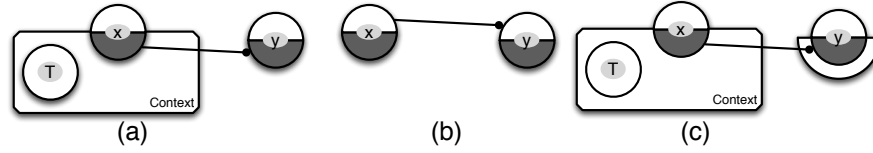

Fig. 6. Constitutive norms and institutional fact.

In our visual language we visualize institutional facts by surrounding with a white border the part of the element which becomes an institutional fact.In Figure 6 illustration **(c)** represent a constitutive rule which consequent $\sim y$ becomes an institutional fact due to the antecedent $\sim x$ being part of the context.

4 Example

In this section we illustrate the visual language, using an example of Cottage Regulations visualized in Figure 7. The example we use is an extension of the *cottage-dog-fence* example used in the literature [11].
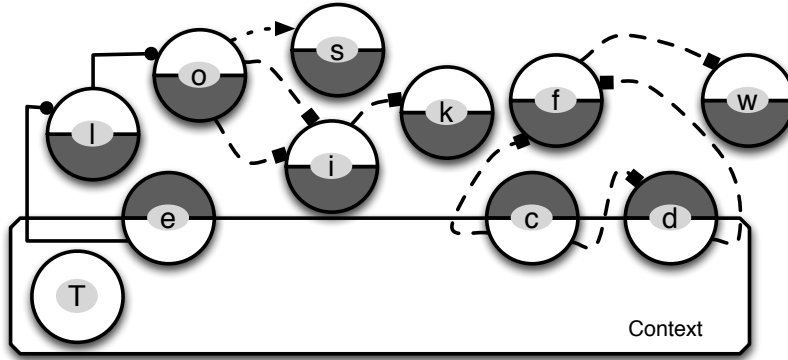

Fig. 7. Cottage Regulations.

4.1 Cottage regulations

The example contains two constitutive norms: "an electronically signed cottage contract (e) counts as a legal cottage contract (l)" and "a legal cottage contract counts as owning the cottage (o)." The regulations prescribe that a person living in a cottage (c) must follow these norms: "dogs (d) are not allowed," "the cottage should not have a fence (f)," "if there is a dog, then there must be a fence," "if there is a fence, then it must be white (w)," "if someone owns a cottage (o), then he (or she) must have an id card (i)," "if someone does not own a cottage, then he (or she) must have an id card" and "if someone has an id card, then he (or she) must keep it with him (or her) (k)." The example presents only one permissive norm: "the owner of a cottage is allowed to sell it (s)."

4.2 Semantics

We define now the semantics for rules activation in our language. To do so we consider a subset of the deontic operations in Definition 4, avoiding $\bigcirc_2$ and $\bigcirc_4$ which require *reasoning by cases*. Reasoning by cases require the introduction of multiple contexts in the visual language which is outside the scope of the present paper. However reasoning by cases represent an important aspect of normative reasoning and tackling this problem is certainly one of our priority as future works.

Considering an arbitrary conditional rule $r = (x, y)$, A a context, I a set of institutional facts, P as a set of positive permissions and O a set of obligations. According to the semantics of the deontic operation $\bigcirc_1$, a rule r is activated if $x \in A \lor x \in I$. Differently if we consider the semantics of $\bigcirc_3$, then the activation semantics changes if r is a regulative or permissive norm. If r is a regulative norm and we are considering $\bigcirc_3$ semantics, then r is activated if $x \in A \lor x \in I \lor x \in P \lor x \in O$. The difference between the two type of rule activation semantics is due to the presence of the transitivity property in $\bigcirc_3$, following the semantics defined in [14].

We consider obligations to be stronger than permissions because if something is obligatory then is implicitly permitted. Due to this reason if an element has to be both obligated and permitted, then the obligation takes over and the element is considered obligatory.

In Figure 8 we describe in details how institutional facts, positive permissions and obligations are derived in the example shown in Figure 7. In Illustration **(a)** in Figure 8 we first apply constitutive norms. The first applicable norm is (e, l) due to having its antecedent enclosed in the context. By applying this norm the element l becomes an institutional fact. According to the activation rule of constitutive norms we can also apply the other norm (l, o) because its antecedent is an institutional fact, changing also o to an institutional fact.

In illustration **(b)** we apply permissive norms. The only permissive norm in the example is (o, s) and because o is an institutional fact we can apply the rule and change s to a positive permission.

In the third illustration **(c)** we apply regulative norms to the example. By looking at the context we can apply three regulative norms: $(c, \sim f), (c, \sim d)$ and (d, f). By applying them we obtain three new obligations: $\sim f, \sim d$ and f. Another regulative norm, (o, i) can be activated because o is an institutional fact, thus i becomes an obligation as well. In the third illustration it is shown what can be derived by not considering the transitivity property.

In the last illustration **(d)** we consider the case where after having reached the state **(c)** we still need to derive obligations and permission due to transitivity. In the example we do not have additional permissive rules to apply, thus we do not generate additional positive permissions. Differently considering regulative norms, two more can be applied due to their antecedents being obligations. These two regulative norms are (i, k) and (f, w) which respectively generate the additional obligations k and w.
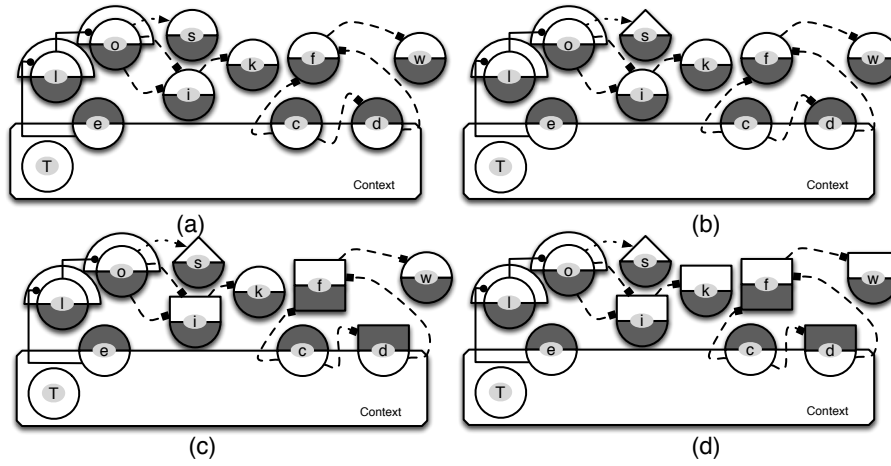

Fig. 8. Deriving institutional facts, permissions and obligations.

5 Patterns

In the present section, some of the recurrent patterns that can be obtained using the proposed visual language to represent normative systems, are analyzed. Using the example described in Section 4 three patterns are highlighted and described: "violation", "contrary to duty" and "dilemma".

5.1 Violation

When dealing with norms, an important aspect that must be taken into account is violations. Differently from constraints which impose impassable boundaries
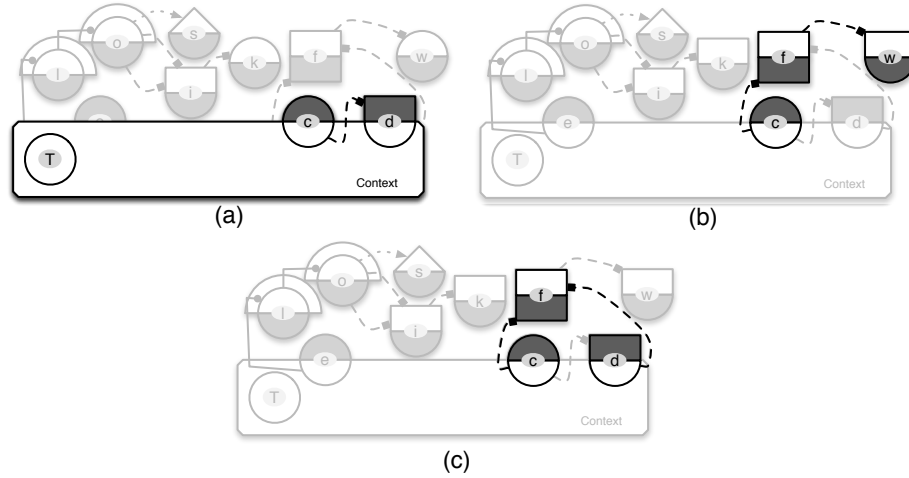
Fig. 9. Violation, Contrary to Duty and Dilemma in the Cottage Regulation example.

in a system, norms can be ignored and violated. For this reason is important in normative systems to recognize violations when they occur.

In Figure 9 illustration (**a**) we highlight a violation that occurs in the cottage regulation example. Considering the highlighted components in the picture we can notice the obligation $\sim d$ has its anti-element enclosed within the context. As previously mentioned we can define a prohibition as the obligation of its anti-element. In this case by having $\sim d$ obliged counts as having d prohibited, thus because d is part of the context means that the norm has been violated.

Generalizing it can be concluded that a violation occurs when an element is obligatory and it's anti-element is part of the context, in other words when an entity is astride the context and the outer element is obligatory.

Another way to violate an obligation is by having an institutional fact as the anti-element of an obligation. This means that one or more elements in the context counts-as having the element that would violate the obligation.

5.2 Contrary to Duty

Contrary to duties are a natural way to cope with violations which cannot be prevented. When a violation occurs, an alternative obligation can be proposed to allow a faulty state to be riconduct to an acceptable state. In Figure 9 illustration (**b**) we highlight the contrary to duty that occurs in the cottage regulations example. We see that obligation $\sim f$ is produced by the element c belonging to the context. Violating this obligation means that its anti-element becomes true. The contrary to duty obligation w is then generated in the case f becomes true, violating $\sim f$ but providing an alternative obligation to follow.

Generally we can identify contrary to duty obligations to a given obligation a by examining which obligation can be generated if the anti-element of a becomes true.

5.3 Dilemma

A dilemma occurs when a normative system generates contrasting obligations. This can occur due to faults inside the normative system or because of two distinct set of regulations interacting on a common situation. Dilemmas are delicate and troublesome positions, but proposing a way to solve them is outside the scope of the present paper. We focus on how they can be recognized in our visual language.

In Figure 9 illustration **(c)** we highlight a dilemma in the cottage regulations example. We see that for entity ⨍, both its elements are obligations derived from c and d enclosed in the context. This position is similar to a "deadlock", because by having an element obligatory, then its anti-element is forbidden. In a dilemma situation both elements of an entity are either obligatory and forbidden.

Dilemmas can be easily spotted in the visual language if an entity has both of its elements drawn as obligations.

6 Conclusion

In this paper we raise the question of how to visualize normative reasoning. We foresee that if normative reasoning is going to be used by a wider community, such a visualization is essential for applications. It can be used to design normative systems, as well as to summarize derivations.

In particular we introduced an abstract approach to visualizing normative reasoning. Since norms are typically represented as rules or conditionals, we represented norms by graphs. Distinguishing as usual regulative, permissive and constitutive norms, we distinguish three kinds of relations in the graph. The actual derivation of generating the obligations, permissions and institutional facts can be visualized as a labeling of the graph. Though this approach fits well input/output logic, it is intended to be a generic notation that can be used with a range of formalisms.

There are other visualizations of graphs too, and it remains to be seen which one works best for normative reasoning. For example, Figure 10 presents a dual visualization that highlights the edges rather than the nodes.

Browsing through the deontic logic literature, a large number of challenges presents itself to the visualization of normative reasoning, such as for example the following ones:

– If there are multiple obligation sets, for example in case of a dilemma, can we visualize them at the same time? How to visualize that we have explosion with both a and $\sim a$? How to visualize defeasibility, where conflicts may be resolved using a conflict resolution principle?
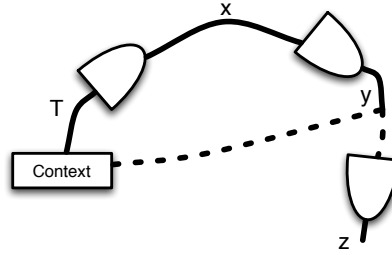
Fig. 10. A dual visualization. $N = \{(x,y), (\sim y, z), (T, x)\}$, $A = \{\top, \sim y\}$

– Time can easily be visualized itself by branching time structures, but more advanced visualization techniques may be needed to combine it with the graphs presented in this paper. In this paper we assumed that the normative system is static, but in many applications norm may change, such that we have to visualize both obligation and norm change in the same figure.
– Agents and actions present yet another challenge. Relations among agents can be visualized abstractly using dependence networks, that represent whether an agent can fulfill the goals of another agent. Norms referring to mental attitudes of agents, like epistemic norms, are a particular challenge here.
– Including the notion of power within the visual language, illustrating which define for an agent which actions are possible or not.

We foresee that some of these challenges like visualizing time and multiple obligation sets, can be solved using more advanced techniques than just diagrams that can be printed, for example using interactive visualizations.

Another topic for further research is experimental validation of the formalism. However, we believe that first we need to define some case studies to further refine the formalism, for example:

– A case study on second life in which the users can design their own normative system for their community, including sanctions.
– A case study on facebook in which people can discuss regulations, for example on privacy.
– The visualization of legal reasoning, where the verdict of a judge is visualized.

We believe that a comparison of visualizations used in reasoning such as Bayesian networks, causal networks, belief revision, dependence networks, CP-nets, and Dung's abstract argumentation systems, may inspire other visualizations of normative reasoning. Moreover, a comparison may also lead to a general theory of visualization of reasoning.

For example, in Dung's abstract theory of argumentation, it can be visualized at the abstract or at the instantiated level. This raises the challenge of visualization of reasoning at various levels of abstraction. For example, the most abstract level may be used to give a general overview of the reasoning, and a

more refined view may illustrate the details. For the theory of abstract normative systems presented here, a natural refinement is the instantiation of the abstract elements with propositional formulas, as in input/output logic. A challenge is thus how to instantiate the graphs with input/output logic derivations.

Acknowledgements

References

1. Roderick M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analyse*, 24:33–36, 1963.
2. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.
3. James William Forrester. Gentle murder, or the adverbial samaritan. *Journal of Philosophy*, 81:193–196, 1984.
4. Gabbay, Horty, van der Meyden, and van der Torre, editors. *Handbook of Normative systems*, volume 1. College Publications, London, To appear a.
5. Bengt Hansson. An analysis of some deontic logics. *Noûs*, 3:373–398, 1969.
6. Lars Lindahl and Jan Odelstad. Normative systems and their revision: An algebraic approach. *Artificial Intelligence and Law*, 11(2-3):81–104, 2003.
7. David Makinson and Leendert van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.
8. David Makinson and Leendert van der Torre. Permissions from an input-output perspective. *Journal of Philosophical Logic*, 32(4):391–416, 2003.
9. Daniel L. Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.
10. Daniel L. Moody and Jos van Hillegersberg. Evaluating the visual syntax of uml :an analysis of the cognitive effectiveness of the uml family of diagrams. In *Software Language Engineering*, volume 5452 of *Lecture Notes in Computer Science*, pages 16–34. . Springer, 2009.
11. Henry Prakken and Marek J. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.
12. James E. Rumbaugh. Notation notes: Principles for choosing notation. *Journal of Object-Oriented Programming*, 8(10):11–14, 1996.
13. John R. Searle. *Speech Acts: an Essay in the Philosophy of Language*. Cambridge University Press, Cambridge (UK), 1969.
14. Silvano Colombo Tosatto, Guido Boella, Leendert van der Torre, and Serena Villata. Abstract normative systems: Semantics and proof theory. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, 2012.
15. Serena Villata, Guido Boella, and Leendert van der Torre. Argumentation patterns. In *Proc. of ArgMAS 2011*, pages 133–150, 2011.