Reified Input/Output logic - a position paper

Livio Robaldo and **Xin Sun**[1]

Abstract. We propose a new approach to formalize obligations and permissions from existing legislation. Specifically, we propose to combine two frameworks: Input/Output logic and the logic of prof. J.R. Hobbs. The former is a well-known framework in normative reasoning. The latter is a neo-Davidsonian wide-coverage first order logic for Natural Language Semantics. We propose to wrap Input/Ouput logic around Hobbs's logic, in order to fill the gap between current logical formalizations of legal text, mostly propositional, and the richness of Natural Language Semantics.

1 Introduction

State-of-the-art systems in legal informatics exploit NLP tools in order to transform, possibly semi-automatically, legal documents into XML standards such as Akoma Ntoso[2], where relevant information are tagged [5] [4]. Although these systems help navigate legislation and retrieve information, their overall usefulness is limited due to their focus on terminological issues while disregarding *semantic* aspects, which allow for legal reasoning.

Deontic Logic (DL) has been used since the 1950s as a formal instrument to model normative reasoning in law [38] [31]. However, subsequent developments in DL adopt an abstract view of law, with a very loose connection with the texts of regulations, which can be addressed with solutions coming from the literature on Natural Language Semantics (NLS). Most current proposals in DL are propositional, while NLS includes a wide range of fine-grained linguistic phenomena that require first-order logic (FOL) formalisms.

We aim at designing a logical framework able to fill the gap between standard (propositional) constructs used in DL and the richness of NLS. Among the logical frameworks (independently) proposed in the literature in NLS and DL respectively, we believe that two of them feature fundamental advantages: (1) the FOL of prof. J.R. Hobbs, designed to model the meaning of NL utterances via *reification*, and (2) Input/Output (I/O) logic, originally proposed in [27] to model deontic normative statements.

Reification is a concept originally introduced by the philosopher D. Davidson in [7]. It allows to move from standard notations in FOL such as '$(give\ a\ b\ c)$', asserting that 'a' gives 'b' to 'c', to another notation in FOL '$(give'\ e\ a\ b\ c)$', where e is the *reification* of the giving action. 'e' is a FOL term denoting the giving event by 'a' of 'b' to 'c'. In line with [2], e is said an "eventuality".

On the other hand, I/O logic is a well-known formalism in DL [9], thanks to its ability to deal with standard problems in DL, e.g., contrary-to-duty reasoning [27] and moral conflicts [32].

This paper presents a possible merging of Hobbs's logic and I/O logic that tries to combine their respective advantages. We restrict our attention to only *obligations* and *permissions*, i.e. the two main kinds of norms [36]. We leave other kinds of norms for future works.

We work on a corpus of EU directives, from which we selected the obligation in (1.a) (*Dir. 98/5/EC*) and the permission in (1.b) (*Dir. 2001/110/EC*). We did not find relevant differences between (1.a-b) and the other norms in the corpus, thus we assume our solution is general enough to cover a representative part of EU legislation.

(1) a. A lawyer who wishes to practise in a Member State other than that in which he obtained his professional qualification shall register with the competent authority in that State.

b. Where baker's honey has been used as an ingredient in a compound foodstuff, the term 'honey' may be used in the product name of the compound food instead of the term 'baker's honey'.

2 Related works

Some approaches in Legal Informatics try to model, in some deontic settings, NL sentences coming from *existing norms*, such as those in (1). The most representative work is perhaps [37]. Other examples may be found in [12] and [1]. Some approaches, e.g. [19], [8], and [15] among others, formalize legal knowledge via Event Calculus [23], a logical language extending reification by introducing special terms and predicates to deal with time points and time periods [10]. A similar account has been investigated by [22] in modal action logic.

To our knowledge, the approach that appears to be closest to the one we are going to propose below is perhaps McCarty's Language for Legal Discourse (LLD) [29]. LLD is strongly drawn on previous studies on NLS, it uses reification, and it aims at modeling existing legal text. [30] shows how it is possible to obtain LLD structures from federal civil cases in the appellate courts in USA via NLP.

However, LLD is very reminiscent of formalisms standardly used in NLS, such as Discourse Representation Theory (DRT) [21], and Minimal Recursion Semantics (MRS) [6]. Those are characterized by a close relation between syntax and semantics, in line with the well-known Montague's *principle of compositionality*[3], a cornerstone of standard formalisms used in NLS.

The principle of compositionality leads to representation based on *embeddings* of subformulae within the logical operators, which establish a *hierarchy* among the predications. For instance, a simple sentence like "John believes that Jack wants to eat an ice cream" could be represented via the following formula (assuming a de-dictio interpretation of the existential quantifier):

[1] University of Luxembourg, Luxembourg, {xin.sun, livio.robaldo}@uni.lu
[2] http://www.akomantoso.org

[3] http://plato.stanford.edu/entries/montague-semantics/#Com

$$believe[\,\texttt{John},\; want(\,\texttt{Jack},\; \exists_x[(iceCream\; x) \wedge (eat\; \texttt{Jack}\; x)]\,]$$

Where $believe$ and $want$ are modal operators taking an individual as first argument and another (embedded) subformula as second argument. In the last formula, the operator $believe$ is hierarchically outscoping the operator $want$, in the sense that the latter occurs within the scope of the former.

Nevertheless, it has been shown by [16], [33], and [34] among others, that such an architecture prevents several available readings in NL, and more complex operators, able to connect the predications across the hierarchy, must be introduced to properly represent them.

For this reason, Hobbs proposed a logic where all formulae are *flat*, i.e. where no hierarchy is established among the predications.

3 Hobbs' logical framework

Prof. J.R. Hobbs defines a wide-coverage first-order logic (FOL) for NLS centered on reification. See [17] and several other earlier publications by the same author[4]. In Hobbs', eventualities may be *possible* or *actual*[5]. This distinction is represented via a predicate $Rexist$ that holds for eventualities really existing in the world. Eventualities may be inserted as parameters of such predicates as $want$, $believe$, etc. Reification can be applied recursively. The fact that "John believes that Jack wants to eat an ice cream" is represented as:

$$\exists_e \exists_{e1} \exists_{e2} \exists_x [\, (Rexist\; e) \wedge (believe'\; e\; \texttt{John}\; e_1) \wedge (want'\; e_1\; \texttt{Jack}\; e_2) \wedge (eat'\; e_2\; \texttt{Jack}\; x) \wedge (iceCream'\; e_3\; x)\,]$$

The crucial feature of Hobbs' logic, which distinguishes it from all other neo-Davidsonian approaches, e.g., LLD, is that all formulae are "flat", in the sense explained above. Specifically, the framework distinguishes between the formulae belonging to the ABox of an ontology from those belonging to its TBox. The ABox only includes conjunctions of atomic predicates asserted on FOL terms. On the other hand, the TBox defines these predicates in terms of the $Rexist$ predicate and standard FOL. All logical operators, e.g., boolean connectives[6], are modeled in this way. For instance, negation is modeled via a predicate not' defined in the TBox as:

(2) For all e and e_1 such that $(not'\; e\; e_1)$ holds, it also holds:

$$(Rexist\; e) \leftrightarrow \neg(Rexist\; e_1)$$

If $(not'\; e\, e_1)$ is true, all what we know is that the individuals e and e_1 are related via the not' predication. But this does not tell us anything about the real existence of either e or e_1. Similarly, and and $imply$ are "conjunctive" and "implicative" relations such that (3) and (4) respectively hold (on the other hand, we omit disjunction).

(3) For all e, e_1, e_2 such that $(and'\; e\; e_1\; e_2)$ holds, it also holds:

$$(Rexist\; e) \leftrightarrow (Rexist\; e_1) \wedge (Rexist\; e_2)$$

(4) For all e, e_1, e_2 such that $(imply'\; e\; e_1\; e_2)$ holds, it also holds:

$$(Rexist\; e) \leftrightarrow ((Rexist\; e_1) \rightarrow (Rexist\; e_2))$$

[4] See manuscripts at http://www.isi.edu/~hobbs/csk.html and http://www.isi.edu/~hobbs/csknowledge-references/csknowledge-references.html.

[5] Other approaches in the literature formalize this distinction in first-order logic, e.g. [3].

[6] See http://www.isi.edu/~hobbs/bgt-logic.text.

Hobbs and his followers implements a fairly large set of predicates for handling composite entities, causality, time, defeasibility, event structure, etc. For instance, [35] proposes a solution to model concessive relations, one of the most trickiest semantic relations occurring in NL, in Hobbs's logic.

The meaning of the predicates is restricted by adding 'axiom schemas'. Space constraints forbid us to illustrate details about all predicates defined by Hobbs. A possible axiom schema for the legal domain is shown in (5). (5) states that all lawyers are humans:

(5) For all e_1, x such that $(lawyer'\; e_1\; x)$ holds, it also holds:

$$\exists e_i \exists e_2[(imply'\; e_i\; e_1\; e_2) \wedge (Rexist\; e_i) \wedge (human'\; e_2\; x)]$$

4 Handling deontic defeasible reasoning in legal interpretation

A major problem in legal informatics concerns the proper interpretation of laws in given situations, which is up to the judges in courts [24]. *Legal interpretation* is a well-studied topic in legal informatics, cf. [25] among others. For instance, in (1.a), to what extent should we think of a lawyer who *wishes* to practise in a Member State different from the one he obtained his qualification? Under a *literal* interpretation of the verb "wishes", which may be taken as its default interpretation, a lawyer who simply *tells* some friends he would like to do so already violates the norm, if he is not registered with the competent authority. On the other hand, a reasonable (pragmatic) interpretation is that the norm is violated only if the non-registered lawyer performs some "formal" action, such as defending someone in court. According to the norm, that action should be blocked and the lawyer must possibly pay a penalty.

So far, few approaches have been proposed to handle multiple legal interpretations in logic. A recent one is [13], where a solution to deal with them in Defeasible Deontic Logic [12] via prioritized defeasible rules is proposed. Priorities are introduced to rank the available interpretations, i.e. to solve potential conflicts among them.

Following [13], we handle multiple legal interpretations via Hobbs's methodology to deal with defeasibility, which is in turn drawn from Circumscriptive Logic [28]. However, we do not claim that our solution features any particular advantage with respect to the one in [13], except the fact that our framework is first-order while Defeasible Deontic Logic is propositional.

The idea is simple and we illustrate it with an example. The fact that every bird flies is represented in FOL as $\forall_x[bird(x) \rightarrow fly(x)]$. In order to render the rule defeasible, we add another predicate $normalBF$ stating that birds fly only if it is "normal" to assume so: $\forall_x[(bird(x) \wedge normalBF(x)) \rightarrow fly(x)]$. Adding that emus are non-flying birds, i.e. $\forall_x[emu(x) \rightarrow (bird(x) \wedge \neg fly(x))]$, does not entail an inconsistency. It entails that $normalBF(x)$ is false for each emu x. In this sense, the latter rule is "stronger" than the former. Alternatively, we may directly assert that emus are not "normal" with respect to the property of flying, i.e. $\forall_x[emu(x) \rightarrow \neg normalBF(x)]$. $normalBF$ must be *assumed* to be true in order to trigger the property of flying on birds.

Different legal interpretations of "wishes" in (1.a) are similarly handled. Let us assume by default that if a lawyer x *says* he will practise in a Member State y, then he really wishes to do it.

(6) For all x, y, e_1, e_2, e_3 such that $(lawyer\; x) \wedge (MS\; y) \wedge (say'\; e_1\; x\; e_2) \wedge (wish'\; e_2\; x\; e_3) \wedge (practice'\; e_3\; x) \wedge (in\; e_3\; y)$ holds, it also holds:

$$\exists e_i[(imply'\; e_i\; e_1\; e_2) \wedge (Rexist\; e_i)]$$

To make (6) defeasible, we add a predicate *normalSP* stating that the entailment is valid only if it is "normal" to assume it:

(7) For all x, y, e_1, e_2, e_3 such that $(lawyer\ x) \wedge (MS\ y) \wedge$ $(say'\ e_1\ x\ e_2) \wedge (wish'\ e_2\ x\ e_3) \wedge (practice'\ e_3\ x) \wedge (in\ e_3\ y)$ holds, it also holds:

$$\exists e_i \exists e_a \exists e_n[(imply'\ e_i\ e_a\ e_2) \wedge (Rexist\ e_i) \wedge (and'\ e_a\ e_1\ e_n) \wedge (normalSP'\ e_n\ e_1)]$$

In (7), the real existence of e_1 is no longer sufficient to entail the one of e_2. In order to enable the entailment, the real existence of e_n is also needed. Now, a judge may reasonably decide that it is *not* normal assuming that a lawyer who says he will practice in a Member State entails that he "wishes" (in the sense of (1.a)) to do so, i.e.:

(8) For all x, y, e_1, e_2, e_3 such that $(lawyer\ x) \wedge (MS\ y) \wedge$ $(say'\ e_1\ x\ e_2) \wedge (wish'\ e_2\ x\ e_3) \wedge (practice'\ e_3\ x) \wedge (in\ e_3\ y)$ holds, it also holds:

$$\exists e^n_n \exists e_n[(not'\ e^n_n\ e_n) \wedge (Rexist\ e^n_n) \wedge (normalSP'\ e_n\ e_1)]$$

From (8), in case a lawyer x simply says he wishes to practice in a Member State y, we infer that e_n does *not* really exist. Thus, it is no longer possible to infer, from (7), whether e_2 really exists or not.

5 Input/Output logic

Input/Output (I/O) logic was introduced in [27]. It originates from the study of conditional norms. I/O logic is a family of logics, just like modal logic is a family of systems K, S4, S5, etc. However, unlike modal logic, which usually uses possible world semantics, I/O logic adopts *operational* semantics: an I/O system is conceived as a "deductive machine", like a black box which produces deontic statements as output, when we feed it factual statements as input.

As explained in [9], operational semantics solves the well-known Jørgensen's dilemma [20], which roughly says that a proper truth-conditional logic of norms is impossible because norms do not carry truth values. According to Jørgensen, typical problems of standard deontic logic arise from its truth-conditional model theory, i.e., possible world semantics. On the other hand, operational semantics straightforwardly allows to deal with contrary-to-duty reasoning, moral conflicts, etc. We address the reader to [26] and [32] among others for further explanations and examples.

Furthermore, I/O logic is one of the few existing frameworks for normative reasoning where also permissions, and not only obligations, have been studied in depth. Most current proposals are not specifically devoted to deal with existing legislation, and so they mostly focus on obligations only. For instance, in [15], devoted to handle business process compliance (BPC), obligations are analyzed in detail, while permissions are mostly neglected, in that the former play a role in BPC more prominent than the latter. The account in [15] has been recently extended to handle permissions in [11].

In [27], four basic I/O logics are defined: $\mathtt{out}_1$, $\mathtt{out}_2$, $\mathtt{out}_3$, and $\mathtt{out}_4$. Let L be standard propositional logic, let O and P be two subsets of L × L, and let A to be a subset of L, i.e. a set of formulae in standard propositional logic. Each pair (a, b) in O is read as "given a, b is obligatory" while each pair (c, d) in P is read as "given c, d is permitted". Pairs in O and P are called "generators" and represent the "deduction machine": whenever one of the left-hand side (LHS) of the pairs is given in input, the corresponding right-hand side (RHS) is given in output.

(9) defines the semantics of $\mathtt{out}_1, \ldots, \mathtt{out}_4$. Cn is the consequence operator of propositional logic; it takes in input a set of formulae A and returns the set corresponding to the transitive closure of all formulae that can be entailed from A. A set of formulas is *complete* if it is either *maximally consistent* or equal to L.

(9)
- $\mathtt{out}_1(O,A)=Cn(O(Cn(A)))$
- $\mathtt{out}_2(O,A)=\bigcap\{Cn(O(V)): A\subseteq V, V$ is complete$\}$
- $\mathtt{out}_3(O,A)=\bigcap\{Cn(O(B)): A\subseteq B = Cn(B) \supseteq O(B)\}$
- $\mathtt{out}_4(O,A)=\bigcap\{Cn(O(V): A\subseteq V \supseteq O(V)), V$ is complete$\}$

In (10), we report the axioms needed to define the I/O systems having the semantics from $\mathtt{out}_1$ to $\mathtt{out}_4$. $\vdash$ is the entailment relation of propositional logic.

(10)
- SI: from (a, x) to (b, x) whenever $b \vdash a$.
- OR: from (a, x) and (b, x) to $(a \vee b, x)$.
- WO: from (a, x) to (a, y) whenever $x \vdash y$.
- AND: from (a, x) and (a, y) to $(a, x \wedge y)$.
- CT: from (a, x) and $(a \wedge x, y)$ to (a, y).

The axioms in (10) constrain the generators belonging to O and P. For instance, CT says that in case two generators (a, x) and $(a \wedge x, y)$ belongs to O, then also the generator (a, y) *must* belong to O.

The derivation system based on SI, WO, and AND is called $\mathtt{deriv}_1$. Adding OR to $\mathtt{deriv}_1$ gives $\mathtt{deriv}_2$. Adding CT to $\mathtt{deriv}_1$ gives $\mathtt{deriv}_3$. The five rules together give $\mathtt{deriv}_4$. Each $\mathtt{deriv}_i$ is sound and complete with respect to $\mathtt{out}_i$ (see [27]).

An example of how the axioms in (10) work in practice is provided below directly on our FOL object logic. As pointed out above, the expressivity of I/O logic, as well as the one of its competitors, e.g., Imperative Logic [14], Prioritized Default Logic [18], and Defeasible Deontic Logic [12] among others, is limited to the *propositional* level. On the other hand, Hobbs's logic, thanks to its formal simplicity, allows to enhance the expressivity of I/O systems to the first-order level with little modifications of the axioms in (10).

6 Combining Input/Output logic and Hobbs's logic

Propositional logic does not have enough expressivity to represent real-world obligations and permissions, such as (1.a-b). Universally quantified variables and constant or functional terms are also needed.

For instance, "a lawyer" and "a Member State" in (1.a) refer to *every* lawyer and *every* Member State. On the other hand, the expression "that in which he obtained his professional qualification" ought to be represented as a function $f_1(x)$ that, given a lawyer x, returns the Member State where he obtained his professional qualification. Similarly, the expression "the competent authority in that State" is represented as a function $f_2(y)$ that, given a Member State y, returns the competent authority in that State. Finally, "the term 'honey' " and "the term 'baker's honey' " in (1.b) correspond to two FOL constants $\mathtt{T}_h$ $\mathtt{T}_{bh}$ respectively, denoting the two English words.

Our formulae are Hobbs's conjunctions of atomic predications, possibly involving FOL variables. Some of those variables will occur both in the LHS and the RHS of an I/O generator, while the others will occur either in the LHS or in the RHS. The variables occurring in both will be universally quantified, while the ones occurring in either one of the two will be existentially quantified. Furthermore, we will require each formula of the object logic to assert exactly one $Rexist$ predicate on the main eventuality. As explained in section 3, the semantics of Hobbs's logic is centered on the $Rexist$ predicate.

We add a single construct to the syntax of the generators: universal quantifiers for binding the variables occurring in both the LHS and the RHS. These quantifiers act as "bridges" between the LHS and the RHS, in order to "carry" individuals from the input to the output. Formally, our generators have the following form, where $LHS(x_1, x_2, \ldots, x_n)$ and $RHS(x_1, x_2, \ldots, x_n)$ are conjunctions of FOL predicates; $x_1, x_2, \ldots, x_n$ are free in LHS and RHS but they are externally bound by universal quantifiers. LHS and RHS will possibly include other existentially quantified variables.

$$\forall_{x_1}\forall_{x_2}\ldots\forall_{x_n}(LHS(x_1, x_2, \ldots, x_n), RHS(x_1, x_2, \ldots, x_n))$$

This architectural choice is motivated by an empirical analysis of the obligations/permissions in our corpus of EU Directives. Norms found in legislation typically hold for all members in a certain set of individuals, e.g. the set of all lawyers. On the other hand, we did not find in our corpus any obligation or permission in the form "If a lawyer exists, then he is obliged to take some actions". This sounds quite intuitive: statements in legislation are typically *universal* assertions, i.e., they do not hold for single specific individuals.

Note that, in any case, as long as formulae are conjunctions of atomic predicates, de re obligations/permissions can be easily dealt with by removing existentials via skolemization. A generator in the form $\exists_x$(LHS(x), $RHS(x)$) can be substituted by (LHS(i), $RHS(i)$), where i is a FOL constant skolemizing $\exists_x$. On the other hand, a generator in the form $\forall_x\exists_y$(LHS(x, y), $RHS(x, y)$) can be substituted by $\forall_x$(LHS(x, $f(x)$), $RHS(x, f(x))$), where f is a FOL function skolemizing $\exists_x$. Existentials occurring in the object logic formulae can be also skolemized. For instance, a generator in the form $\forall_x(\exists_y LHS(x, y)$, $RHS(x)$) can be substituted by $\forall_x$(LHS(x, $f(x)$), $RHS(x)$), where f is a FOL function skolemizing $\exists_x$.

Similarly, it must be observed that, in finite domains, universal quantifiers are just a compact way to refer to all individuals in the universe. We obtain an equivalent set of generators by substituting the universally quantified variables with all constants referring each to an individual in the universe. For instance, assuming the universe includes the individuals a, b, c only, the generator $\forall_x$(LHS(x), $RHS(x)$) is equivalent to the set of generators (LHS(a), $RHS(a)$), (LHS(b), $RHS(b)$), and (LHS(c), $RHS(c)$).

6.1 Generalizing Input/Output logic axioms

We have proposed above to integrate Hobbs's logic within I/O generators by simply adding wide-scope universal quantifiers to the syntax of the generators, in order to create a "bridge" for "carrying" the FOL terms matching the LHS to the output. Also the axioms in (10) need to be generalized accordingly. This section shows the generalization of the axiom CT. The generalization of the other axioms is similar and it is left to the reader as an exercise. CT is generalized as in (11).

(11) from: $\forall_{x_1}\ldots\forall_{x_n}($
$\exists_{e_{11}}\exists_{y_{11}}\ldots\exists_{y_{1i}}[(Rexist\, e_{11}) \wedge (\Psi'_1\, e_{11}\, y_{11}\ldots y_{1i}\, x_1\ldots x_n)]$,
$\exists_{e_{21}}\exists_{y_{21}}\ldots\exists_{y_{2j}}[(Rexist\, e_{21}) \wedge (\Psi'_2\, e_{21}\, y_{21}\ldots y_{2j}\, x_1\ldots x_n)])$
and: $\forall_{x_1}\ldots\forall_{x_n}(\exists_e\exists_{e_{11}}\exists_{y_{11}}\ldots\exists_{y_{1i}}\exists_{e_{21}}\exists_{y_{21}}\ldots\exists_{y_{2j}}[$
$(Rexist\, e) \wedge (and'\, e\, e_{11}\, e_{21}) \wedge (\Psi'_1\, e_{11}\, y_{11}\ldots y_{1i}\, x_1\ldots x_n) \wedge$
$(\Psi'_2\, e_{21}\, y_{21}\ldots y_{2j}\, x_1\ldots x_n)]$,
$\exists_{e_{31}}\exists_{y_{31}}\ldots\exists_{y_{3k}}[(Rexist\, e_{31}) \wedge (\Psi'_3\, e_{31}\, y_{31}\ldots y_{3k}\, x_1\ldots x_n)])$
to: $\forall_{x_1}\ldots\forall_{x_n}($
$\exists_{e_{11}}\exists_{y_{11}}\ldots\exists_{y_{1i}}[(Rexist\, e_{11}) \wedge (\Psi'_1\, e_{11}\, y_{11}\ldots y_{1i}\, x_1\ldots x_n)]$,
$\exists_{e_{31}}\exists_{y_{31}}\ldots\exists_{y_{3k}}[(Rexist\, e_{31}) \wedge (\Psi'_3\, e_{31}\, y_{31}\ldots y_{3k}\, x_1\ldots x_n)])$

An example is: given "Every lawyer is obliged to run" and "Every lawyer who runs is obliged to wear a red hat", formalized in (12):

(12) $\forall_x(\ \exists_{e_{11}}[(Rexist\, e_{11}) \wedge (lawyer'\, e_{11}\, x)]$,
$\exists_{e_{21}}[(Rexist\, e_{21}) \wedge (run'\, e_{21}\, x)]\ )$

$\forall_x(\ \exists_e\exists_{e_{11}}\exists_{e_{21}}[(Rexist\, e) \wedge (and'\, e\, e_{11}\, e_{21}) \wedge$
$(lawyer'\, e_{11}\, x) \wedge (run'\, e_{21}\, x)]$,
$\exists_{e_{31}}[(Rexist\, e_{31}) \wedge (wearRedHat'\, e_{31}\, x)]\ )$

in case the I/O system includes the axiom in (11), O must include (13), which refers to "Every lawyer is obliged to wear a red hat".

(13) $\forall_x(\ \exists_{e_{11}}[(Rexist\, e_{11}) \wedge (lawyer'\, e_{11}\, x)]$,
$\exists_{e_{31}}[(Rexist\, e_{31}) \wedge (wearRedHat'\, e_{21}\, x)])$

6.2 Formalizing the examples in (1)

We have now all the ingredients for representing (1.a-b). In a normative Input/Output system N=(O,P), the former is inserted in O while the latter is inserted in P. The formula representing (1.a) is:

(14) $\forall_x\forall_y(\ \exists_{e_1}\exists_{e_2}[(Rexist\, e_1) \wedge (lawyer\ x) \wedge (MS\ y) \wedge$
$(wish'\ e_1\ x\ e_2) \wedge (practice'\ e_2\ x) \wedge (in\ \ e_2\ y) \wedge diffFrom(y\ f_1(x))]$,
$\exists_{e_3}[(Rexist\, e_3) \wedge (register'\ \ e_3\ \ x) \wedge (at\ e_3\ \ f_2(y))]\ )$

As discussed in section 4, the predicate *wish*, as well as any other predicate, may be subject to different legal interpretations, which may be asserted in the knowledge base via the mechanism used in Hobbs's to handle defeasibility.

The permission in (1.b) is similarly formalized as in (15).

(15) $\forall_y(\exists_x\exists_{e_1}[(Rexist\, e_1) \wedge (ingrOf'\ e_1\ x\ y) \wedge$
$(bakerHoney\ \ x) \wedge (foodStuff\ \ y)]$,
$\exists_{e_2}[(Rexist\, e_2) \wedge (substitute'\ \ e_2\ \ \mathtt{T}_h\ \mathtt{T}_{bh}) \wedge (in\ \ e_2\ f_3(y))]$

Note that the variable x occurs in the LHS only, thus it is existentially quantified. The formula in (15) reads as follows: for each compound foodstuff y for which it is "true" (in the sense that it really exists in the current world) the fact that one of its ingredients is baker's honey, then it is permitted that, in the current world, also the fact that the term 'honey' is substituted by the term 'baker's honey' in the product name of y really exist.

ACKNOWLEDGEMENTS

Livio Robaldo has received funding from the European Unions H2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 661007 for the project "ProLeMAS: PROcessing LEgal language in normative Multi-Agent Systems". Xin Sun has received funding from the European Union's H2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 690974 for the project "MIREL: MIning and REasoning with Legal texts".

REFERENCES

[1] *Logic in the theory and practice of lawmaking*, eds., M. Araszkiewicz and K. Pleszka, Springer, 2015.
[2] E. Bach, 'On time, tense, and aspect: An essay in english metaphysics', in *Radical Pragmatics*, ed., P. Cole, Academic Press, New York, (1981).

[3] P. Blackburn and J. van Benthem, 'Modal logic: A semantic perspective', in *Handbook of Modal Logic*, eds., P. Blackburn, J. van Benthem, and F. Wolter, Elsevier, (2007).

[4] G. Boella, L. Di Caro, and L. Robaldo, 'Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines', in *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pp. 218–225. Springer, (2013).

[5] G. Boella, L. Di Caro, A. Ruggeri, and L. Robaldo, 'Learning from syntax generalizations for automatic semantic annotation', *Journal of Intelligent Information Systems*, **43**(2), 231–246, (2014).

[6] A. Copestake, D. Flickinger, and I.A. Sag, 'Minimal Recursion Semantics. An introduction', *Journal of Research on Language and Computation.*, **2**(3), (2005).

[7] D. Davidson, 'The logical form of action sentences', in *The Logic of Decision and Action*, ed., N. Rescher, Univ. of Pittsburgh Press, (1967).

[8] A. Farrell, M. Sergot, M. Salle, and C. Bartolini, 'Using the event calculus for tracking the normative state of contracts', *International Journal of Cooperative Information Systems*, **14**, 99–129, (2005).

[9] D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. (eds.) van der Torre, *Handbook of Deontic Logic and Normative Systems*, College Publications, 2013.

[10] Antony Galton, 'Operators vs. arguments: The ins and outs of reification', *Synthese*, **150**(3), 415–441, (2006).

[11] G. Governatori and M. Hashmi, 'Permissions in deontic event-calculus', in *International Conference on Legal Knowledge and Information Systems (Jurix)*, pp. 181–182, Braga, Portugal, (2015).

[12] G. Governatori, F. Olivieri, A. Rotolo, and S. Scannapieco, 'Computing strong and weak permissions in defeasible logic', *Journal of Philosophical Logic*, **6**(42), (2013).

[13] G. Governatori, A. Rotolo, and G. Sartor, 'Deontic defeasible reasoning in legal interpretation', in *The 15th International Conference on Artificial Intelligence & Law*, ed., K. Atkinson, San Diego, (2015).

[14] Jörg Hansen, 'Prioritized conditional imperatives: problems and a new proposal', *Autonomous Agents and Multi-Agent Systems*, **17**(1), (2008).

[15] M. Hashmi, G. Governatori, and M. Wynn, 'Modeling obligations with event-calculus', in *Rules on the Web. From Theory to Applications*, eds., A. Bikakis, P. Fodor, and D. Roman, volume 8620 of *Lecture Notes in Computer Science*, 296–310, Springer International Publishing, (2014).

[16] J. R. Hobbs, 'Deep lexical semantics', in *Proc. of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008)*, Haifa, Israel, (2008).

[17] J.R. Hobbs, 'The logical notation: Ontological promiscuity', in *Chap. 2 of Discourse and Inference*, (1998). Available at http://www.isi.edu/∼hobbs/disinf-tc.html.

[18] John Horty, *Reasons as Defaults*, Oxford University Press, 2012.

[19] Andrew J.I. Jones and Steven Orla Kimbrough, *A Note on Modelling Speech Acts as Signalling Conventions*, 325–342, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[20] Jorgen Jørgensen, 'Imperatives and logic', *Erkenntnis*, **7**, (1937).

[21] H. Kamp and U. Reyle, *From Discourse to Logic: an introduction to model-theoretic semantics, formal logic and Discourse Representation Theory*, Kluwer Academic Publishers, Dordrecht, 1993.

[22] Samit Khosla and T. S. E. Maibaum, 'The prescription and description of state based systems', in *Temporal Logic in Specification, Altrincham, UK, April 8-10, 1987, Proceedings*, eds., Behnam Banieqbal, Howard Barringer, and Amir Pnueli, volume 398 of *Lecture Notes in Computer Science*, pp. 243–294. Springer, (1987).

[23] R Kowalski and M Sergot, 'A logic-based calculus of events', *New Generation Computing*, **4**(1), 67–95, (1986).

[24] Doris Liebwald, 'Vagueness in law: A stimulus for 'artificial intelligence & law'', in *Proc. of the 14th International Conference on Artificial Intelligence and Law*, ICAIL '13, pp. 207–211, (2013).

[25] N. MacCormick and R.S. Summers, *Interpreting Statutes: A Comparative Study*, Applied legal philosophy, Dartmouth, 1991.

[26] David Makinson and Leendert van der Torre, 'Constraints for input/output logics', *Journal of Philosophical Logic*, **30**(2), (2001).

[27] David Makinson and Leendert W. N. van der Torre, 'Input/output logics', *Journal of Philosophical Logic*, **29**(4), 383–408, (2000).

[28] J. McCarthy, 'Circumscription: A form of nonmonotonic reasoning', *Artificial Intelligence*, (13), 27–39, (1980).

[29] L. T. McCarty, 'Ownership: A case study in the representation of legal concepts', *Artificial Intelligence and Law*, **10**(1-3), 135–161, (2002).

[30] L. T. McCarty, 'Deep semantic interpretations of legal texts', in *The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 4-8, 2007, Stanford Law School, Stanford, California, USA*, pp. 217–224, (2007).

[31] J.J. Meyer and R.J. Wieringa, *Deontic Logic in Computer Science: Normative system specification.*, John Wiley and sons Ltd, 1993.

[32] X. Parent, 'Moral particularism in the light of deontic logic', *Artificial Intelligence and Law*, **19**(2-3), 75–98, (2011).

[33] L. Robaldo, 'Interpretation and inference with maximal referential terms.', *The Journal of Computer and System Sciences*, **76(5)**, (2010).

[34] L. Robaldo, 'Conservativity: a necessary property for the maximization of witness sets.', *The Logic Journal of the IGPL*, **21(5)**, 853–878, (2013).

[35] L. Robaldo and E. Miltsakaki, 'Corpus-driven semantics of concession: Where do expectations come from?', *Dialogue&Discourse*, **5**(1), (2014).

[36] G. Sartor and E. Pattaro, *Legal Reasoning: A Cognitive Approach to the Law*, Treatise of legal philosophy and general jurisprudence, 2005.

[37] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory, 'The british nationality act as a logic program', *Communications of the ACM*, **29**(5), (1986).

[38] G. H. von Wright, *An Essay in Deontic Logic and the General Theory of Action*, Acta Philosophica Fennica, Fasc. 21, North-Holland, 1968.