One License to Compose Them All

A Deontic Logic Approach to Data Licensing on the Web of Data

Guido Governatori[1*], Antonino Rotolo[2], Serena Villata[3**], and Fabien Gandon[3]

[1] NICTA Queensland Research Laboratory
[2] University of Bologna
[3] INRIA Sophia Antipolis

Abstract In the domain of Linked Open Data a need is emerging for developing automated frameworks able to generate the licensing terms associated to data coming from heterogeneous distributed sources. This paper proposes and evaluates a deontic logic semantics which allows us to define the deontic components of the licenses, i.e., permissions, obligations, and prohibitions, and generate a composite license compliant with the licensing items of the composed different licenses. Some heuristics are proposed to support the data publisher in choosing the licenses composition strategy which better suits her needs w.r.t. the data she is publishing.

1 Introduction

Following the Open Data movement[1], several data hubs are being created by public bodies from single cities through to supra national organizations like the European Union with the final aim to improve the transparency and efficiency of such public bodies and organizations. In this context, the data is openly published on the Web using different data models (e.g., RDF, schema.org, CSV). However, even if this movement is receiving more attention in the last years, still much more effort is required to publish open data on the Web, possibly in a machine-readable format in such a way that data could be interlinked, supporting the growth of the Web of Data [3,13]. One open problem in this context is *quality assessment* with a particular attention to provenance information [12]. More precisely, part of the self-description of the data consists in the licensing terms which specify the admitted use and re-use of the data by third parties. This issue is relevant both for *i)* Linked Data publication as underlined in the "7 Best Practices for Producing Linked Data"[2] where it is required to specify an appropriate license for the data, and *ii)* Open Data publication since the possibility to express constraints on the reuse of the data would encourage the publication of more open data. In this paper, we answer the research question: *How to express the licensing terms associated to*

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

** The author acknowledges support of the DataLift Project ANR-10-CORD-09 founded by the French National Research Agency.

[1] `http://www.w3.org/TR/gov-data/`

[2] `http://www.w3.org/2011/gld/wiki/Linked_Data_Cookbook`

data coming from heterogeneous distributed sources? This research question could be answered by linking the datasets to normative documents describing the licenses under which they are released. However, this solution is far from the Web of Data philosophy where the meta-information about the datasets should be expressed both in a human and a machine-readable format to allow further reasoning steps. Thus the research question breaks down into the following subquestions: *i)* How to express the deontic component of the licensing terms in a machine-readable format?, and *ii)* How to compose in a compliant and automated way the licensing terms associated to a set of heterogeneous data to produce a single composite license?

First, we introduce a lightweight vocabulary called `l4lod`[3] (Licenses for Linked Open Data) which is composed by the three main deontic components, i.e., Obligations, Permissions and Prohibitions, and provides an alignment with the other licenses vocabularies. It is used to express the machine-readable composite license.

Second, we rely on the deontic logic paradigm [24] to address the problem of *reconciling* a set of licenses associated to heterogeneous datasets whose information items are returned together for consumption, e.g., resulting from a single SPARQL query over distributed datasets released under different licenses. Assuming that these datasets provide the consumer with their own licensing terms, we propose and evaluate a deontic logic semantics which automatically returns to the consumer a so called *composite license* which is compliant with the normative semantics of each single license composing it.

The rationale of this work is to support both consumers and publishers of Linked Open Data. On the one hand, as a consumer it is fundamental to know the kind of operations you are permitted to perform on the data to avoid data misuses. On the other hand, we support the publisher to decide which heuristics better suits her needs about the composition of the licenses associated to her data, e.g., the composite license with less obligations and more permissions is preferred to the others.

The reminder of the paper is as follows. Section 2 starts with an analysis of the Linked Data cloud from the licenses point of view and presents the `l4lod` vocabulary. In Section 3, we present our deontic logic to represent and reason over the licensing terms together with the heuristics to guide licenses composition. In Section 4, we evaluate our approach using the SPINdle logic reasoner. In Section 5 we present the existing research, and we compare it with the proposed approach.

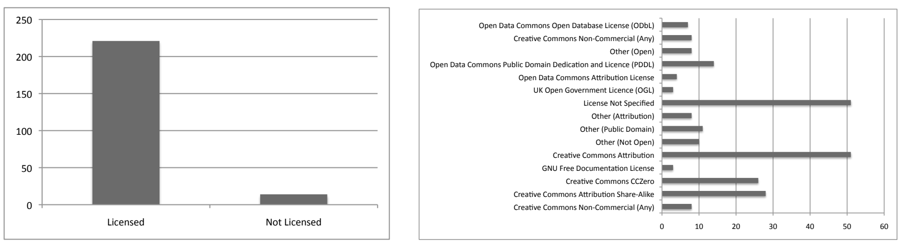
2 Licenses for Linked Open Data

The first issue to be addressed with respect to the use of licenses in Linked Open Data (LOD) is to understand how many datasets of the LOD cloud[4] are actually licensed, and, at a later stage, which are the more popular licenses adopted in those datasets. In order to perform such analysis, we crawled the LOD cloud[5] with a total of 235 datasets considered. The results of this analysis are as follows:

[3] `http://ns.inria.fr/l4lod/`
[4] `http://lod-cloud.net/`
[5] The Data Hub: Linking Open Data Cloud. Retrieved May 02, 2013 (UTC).
`http://datahub.io/group/lodcloud`

Licensed-Not Licensed 221 datasets out of 235 are licensed in some way (Figure 1(a)). The licensing terms are often reported in the VOID meta-data[6] using the `dcterms:license` or the `dcterms:rights` properties of Dublin Core[7]. The license is not usually explicited in a machine-readable format, i.e., the URI of the license is given, but it brings to the human-readable version only (around 95%).

Licenses distribution The most adopted license is Creative Commons Attributions (CC-BY)[8] (51 out of 221 datasets), and the Creative Commons (CC) licenses [1] in general represent the 51% of the licenses used on the LOD cloud (Figure 1(b)). Other popular licenses are Open Data Commons (ODC) ones[9] [18] (11%), and other licenses from specific institutions (18%)[10].



(a) (b)

Figure 1. Surveying the Linked Data Cloud: licensed data statistics.

We introduce the `l4lod` lightweight vocabulary (Licenses for Linked Open Data) which is used to collect and align existing vocabularies which specify with different granularity levels the licensing terms associated to the data. `l4lod` is adopted in our framework as reference vocabulary to specify in a machine-readable format the licensing terms associated to the composite license we automatically generate. Moreover, starting by the observation that not all licensed works are *creative works* [13], `l4lod` may be used to specify the deontic components for those licenses outside CC, like for instance ODC licenses, and the Open Government License (OGL) [11], as through the Open Digital Rights Language (ODRL) vocabulary[12]. The fine grained specification of licensing terms in a machine-readable format is the goal of the ODRL vocabulary, while the aim of `l4lod` is to describe the composite license at the level of its basic deontic components. We are currently investigating how to use the ODRL vocabulary to address the `l4lod` requirements as an ODRL Profile.

[6] `http://www.w3.org/TR/void/`
[7] `http://purl.org/dc/terms/`
[8] `http://creativecommons.org/licenses/by/3.0/`
[9] `http://opendatacommons.org/licenses/`
[10] LOD cloud highlighting licenses distribution available at `http://ns.inria.fr/l4lod/`
[11] `http://www.nationalarchives.gov.uk/doc/open-government-licence/`
[12] `http://www.w3.org/community/odrl/two/model/`

We define the class License which is equivalent to cc:License[13] and limo:LicenseModel[14], and three basic deontic properties which are respectively permits, prohibits, and obliges. These properties connect each license with its own elements: Reproduction, Derivative, Distribution, Sharing, Using, CommercialExpl, Publishing (for *Permissions*), Attribution, ShareAlike, Citation (for *Obligations*), and NoCommercial, NoDerivative (for *Prohibitions*). The vocabulary does not provide an exhaustive set of properties for licenses definition. Implementations are free to extend l4lod to add further elements. In the vocabulary, we distinguish between facts (i.e., rights as in the class License) and their representation. That is why we introduce the licensingTerms property to connect the license to its human-readable counterpart (domain dc:LicenseDocument). Further distinctions, e.g., among facts/information, collections of facts, are out of the scope of this vocabulary and they are carried out by other vocabularies (e.g., ODRL).

The vocabulary considers, among others, the alignment with the following vocabularies: the CC vocabulary, the ODRL vocabulary[15], the LiMo vocabulary, the Dublin Core vocabulary, the Waiver vocabulary[16], the Description of a Project vocabulary (doap)[17], the Ontology Metadata vocabulary (omv)[18], the Data Dictionary for Preservation Metadata (premis)[19], the Vocabulary Of Attribution and Governance (voag)[20].

3 Defeasible deontic logic for licenses composition

We propose an extension of Defeasible Logic, revising earlier works [8,9], to handle license composition. Dealing with this issue requires reasoning about two components:

Factual and ontology component: the first component is meant to describe the facts with respect to which Web of Data licenses are applied as well as the ontology of concepts involved by licenses (thus modeling, e.g., concept inclusion);

Deontic component: the second component aims at capturing the deontic aspects of Web of Data licenses, thus offering mechanisms for reasoning about obligations, prohibitions, and permissions in force in each license, and in their composition.

In this paper, we basically focus on the deontic component, even though, for the sake of completeness, we illustrate the proposed method by also handling, in standard Defeasible Logic, the factual and ontology component, as done in [4]. However, standard Defeasible Logic is just an option, and the factual and ontology component can be handled in any other suitable logic and by resorting to a separate reasoner. Also, notice that we assume that all licenses share a same ontology, or the ontologies are aligned.

[13] http://creativecommons.org/ns
[14] http://purl.org/LiMo/0.1
[15] http://w3.org/ns/odrl/2/
[16] http://vocab.org/waiver/terms/.html
[17] http://usefulinc.com/ns/doap
[18] http://omv2.sourceforge.net/index.html
[19] http://bit.ly/premisOntology
[20] http://voag.linkedmodel.org/schema/voag

The formal language of the logic is rule-based. Literals can be plain, such as $p, q, r \ldots$, or modal, such Op (obligatory), Pp (permitted), and Fp (forbidden/prohibited). Ontology rules work as regular Defeasible Logic rules for deriving plain literals, while the logic of deontic rules provide a constructive account of the basic deontic modalities (obligation, prohibition, and permission). However, while we assume that all licenses share a same ontology, the purpose of the formalism is mainly to establish the conditions to derive *different* deontic conclusions from *different* licenses, and check whether they are compatible so that they can be attributed to a composite license. Hence, we need to keep track of how these deontic conclusions are obtained. To this purpose, deontic rules (and, as we will see, their conclusions) are parametrized by labels referring to licenses.

An ontology rule such as $a_1, \ldots, a_n \Rightarrow b$ supports the conclusion of b, given $a_1, \ldots, a_n$, and so it states that, from the viewpoint of any license any instance enjoying $a_1, \ldots, a_n$ is also an instance of b. On the contrary, rules as $a, \mathrm{O}b \Rightarrow_{\mathrm{O}}^{l_2} p$ state that, if a is the case and b is obligatory, then Op holds in the perspective of license l_2, i.e., p is obligatory for l_2.

The proof theory we propose aims at offering an efficient method for reasoning about the deontic component of each license and, given that method, for combining different licenses, checking their compatibility, and establishing what deontic conclusions can be drawn from the composite license. In other words, if $l_c = l_1 \odot \cdots \odot l_n$ is the composite license obtained from $l_1, \ldots, l_n$, the conclusions derived in the logic for $l_1, \ldots, l_n$ are also used to establish those that hold in l_c.

The reader may argue about the choice of defeasible deontic logics. A simpler approach would be to foster the adoption of standardized licenses and assign them a URI. Then, a basic URI comparison can trigger the allowed/appropriate usages of the data. However, even if we support such kind of standardization, we believe that it is far from the present situation where different licenses are used on the Web, from the basic purpose licenses up to the national ones. Dealing with licenses composition requires reasoning about all deontic provisions, handling and solving normative conflicts arising from deontically incompatible licenses, and exceptions. A few formalisms can do that. Defeasible deontic logic is one of the best candidates, as all aspects are managed in an efficient and computationally tractable way.

3.1 Formal language and basic concepts

The basic language is defined as follows. Let $\mathrm{Lic} = \{l_1, l_2, \ldots, l_n\}$ be a finite set of licenses. Given a set PROP of *propositional atoms*, the set of *literals* Lit is the set of such atoms and their negation; as a convention, if q is a literal, $\sim q$ denotes the complementary literal (if q is a positive literal p then $\sim q$ is $\neg p$; and if q is $\neg p$, then $\sim q$ is p). Let us denote with $\mathrm{MOD} = \{\mathrm{O}, \mathrm{P}, \mathrm{F}\}$ the set of basic deontic modalities. The set ModLit of modal literals is defined as follows: i) if $X \in \mathrm{MOD}$ and $l \in \mathrm{Lit}$, then Xl and $\neg Xl$ are modal literals, ii) nothing else is a modal literal.

Let Lbl be a set of arbitrary labels. Every rule is of the type $r : A(r) \hookrightarrow_Y^x C(r)$, where

1. $r \in \mathrm{Lbl}$ is the name of the rule;
2. $A(r) = \{a_1, \ldots, a_n\}$, the *antecedent* (or *body*) of the rule, is a finite set denoting the premises of the rule. If r is an ontology rule, then each a_i, $1 \leq i \leq n$, belongs to Lit, otherwise it belongs to $\mathrm{Lit} \cup \mathrm{ModLit}$;

3. $\hookrightarrow \in \{\rightarrow, \Rightarrow, \rightsquigarrow\}$ denotes the type of the rule;
4. if r is a deontic rule, $Y = \mathrm{O}$ represents the type of conclusion obtained[21]; otherwise (for ontology rules), $Y \in \emptyset$;
5. if r is a deontic rule, $x \in \mathrm{Lic}$ indicates to which license the rule refers to; otherwise (for ontology rules), $x \in \emptyset$;
6. $C(r) = b \in \mathrm{Lit}$ is the *consequent* (or *head*) of the rule.

The intuition behind the different arrows is the following. *Strict rules* have the form $a_1, \ldots, a_n \rightarrow^x_Y b$. *Defeasible rules* have the form $a_1, \ldots, a_n \Rightarrow^x_Y b$. A rule of the form $a_1, \ldots, a_n \rightsquigarrow^x_Y b$ is a *defeater*. Analogously, for ontology rules, where arrows do not have superscripts and subscripts. The three types of rules establish the strength of the relationship. Strict rules provide the strongest connection between a set of premises and their conclusion: whenever the premises are deemed as indisputable so is the conclusion. Defeasible rules allow to derive the conclusion unless there is evidence for its contrary. Finally, defeaters suggest that there is a connection between its premises and the conclusion not strong enough to warrant the conclusion on its own, but such that it can be used to defeat rules for the opposite conclusion.

A multi-license theory is the knowledge base which is used to reason about the applicability of license rules under consideration.

Definition 1. *A* multi-license theory *is a structure* $D = (F, L, R^c, \{R^{\mathrm{O}^l}\}_{l \in \mathrm{Lic}}, \succ)$*, where*

- $F \subseteq \mathrm{Lit} \cup \mathrm{ModLit}$ *is a finite set of facts;*
- $L \subseteq \mathrm{Lic}$ *is a finite set of licenses;*
- R^c *is a finite set of ontology rules;*
- $\{R^{\mathrm{O}^l}\}_{l \in \mathrm{Lic}}$ *is finite family of sets of obligation rules;*
- $\succ$ *is an acyclic relation (called superiority relation) defined over* $(R^c \times R^c) \cup (R^{\mathrm{O}^l} \times R^{\mathrm{O}^{l'}})$*, where* $R^{\mathrm{O}^l}, R^{\mathrm{O}^{l'}} \in \{R^{\mathrm{O}^l}\}_{l \in \mathrm{Lic}}$[22].

$R[b]$ and $R^X[b]$ with $X \in \{c, \mathrm{O}^l | l \in \mathrm{Lic}\}$ denote the set of all rules whose consequent is b and of all rules (of type X). Given a set of rules R the sets R_{s}, R_{sd}, and R_{dft} denote, respectively, the subsets of R of strict rules, defeasible rules, and defeaters.

3.2 Proof theory

A *proof* P of length n is a finite sequence $P(1), \ldots, P(n)$ of tagged literals of the type $+\Delta^X q$, $-\Delta^X q$, $+\partial^X q$ and $-\partial^X q$, where $X \in \{c, Y^l | l \in \mathrm{Lic}, Y \in \mathrm{MOD}\}$. The proof conditions below define the logical meaning of such tagged literals. As a conventional notation, $P(1...i)$ denotes the initial part of the sequence P of length i. Given a multi-license theory D, $+\Delta^X q$ means that literal q is provable in D with the mode X using only facts and strict rules, $-\Delta^X q$ that it has been proved in D that q is not definitely provable in D with the mode X, $+\partial^X q$ that q is defeasibly provable in D with the mode X, and $-\partial^X q$ that it has been proved in D that q is not defeasibly provable in D with the mode X[23].

[21] We will see why we do not need rules for prohibitions and permissions.
[22] Notice that we may have that $l = l'$.
[23] As we will see, we shall adopt a reading of permissions according to which they can only be defeasible. Hence, we will not define the cases $\pm\Delta^{Y^l} q$ where $Y = \mathrm{P}$.

Given $\# \in \{\Delta, \partial\}$, $P = P(1), \ldots, P(n)$ *is a proof for* p in D for the license l iff $P(n) = +\#^l p$ when $p \in \text{Lit}$, $P(n) = +\#^{X^l} q$ when $p = Xq \in \text{ModLit}$, and $P(n) = -\#^{Y^l} q$ when $p = \neg Yq \in \text{ModLit}$.

The proof conditions aim at determining what conclusions can be obtained within composite licenses by using the source licenses. Three heuristics have been proposed for this purpose [6,23]:

- **OR-composition:** if at least one of the licenses owns a clause then also l_c owns it;
- **AND-composition:** if all the licenses own a clause then also l_c owns it;
- **Constraining-value:** the most constraining clause among those offered by the single licenses is included in l_c.

In this paper, we concentrate on deontic effects of licenses, thus working on the obligations, prohibitions, permissions entailed by the composition of a given set of licenses (instead of the composition of the clauses). Also, since the constraining-value heuristics requires to fully model the idea of concept inclusion (thus working also on the ontology part; see discussion in [21]), here we focus the first two heuristics, reframed as:

- **OR-composition:** l_c entails a deontic effect if there is at least one license that entails such effect (and no license prevents it).
- **AND-composition:** l_c entails a deontic effect if all licenses entail it.

In the next sections, we will show by means of examples, how the AND- and OR-heuristics operate in the logic, including the derived conclusions.

Some notational conventions and concepts that we will use throughout the remainder of this section: *i)* let $l_c = l_1 \odot \cdots \odot l_n$ be any composite license that can be obtained from the set of licenses $L_c = \{l_1, \ldots, l_n\} \subseteq L$; *ii)* let $X, Y \in \text{MOD}$.

As usual with Defeasible Logic, we have proof conditions for the monotonic part of the theory (proofs for the tagged literals $\pm\Delta^Y p$) and for the non-monotonic part (proofs for the tagged literals $\pm\partial^Y p$). To check licenses' compatibility and compose them means to apply the proof conditions of the logic to a multi-license where the set of licenses is $L = L_c$. Since the proof theory for the ontology component ($\pm\Delta^c p$ and $\pm\partial^c p$) is the one for standard Defeasible Logic we will omit it and refer the reader to [2]. For $\# \in \{\Delta, \partial\}$ and $Y \in \{\text{O}, \text{P}, \text{F}\}$, notice that conditions governing conclusions for the composite license l_c and for any each license l_i interplay recursively: indeed, we may use a conclusion for l_c to fire a rule in l_i.

3.3 Provability in each license

Definite Provability The definitions below for Δ describe just forward (monotonic) chaining of strict rules.

Obligation Definite Provability

$+\Delta^{\text{O}^{l_i}}$: If $P(n+1) = +\Delta^{\text{O}^{l_i}} q$ then,
(1) $\text{O}q \in F$ or
(2) $\exists r \in R_s^{\text{O}^{l_i}}[q]$:
$\forall a, Xb, \neg Yd \in A(r)$:
$+\Delta^c a$, $+\Delta^{X^{l_c}} b$, $-\Delta^{Y^{l_c}} d \in P(1..n)$

$-\Delta^{\text{O}^{l_i}}$: If $P(n+1) = -\Delta^{\text{O}^{l_i}} q$ then
(1) $\text{O}q \notin F$ and
(2) $\forall r \in R_s^{\text{O}^{l_i}}[q]$:
$\exists a \in A(r)$: $-\Delta^c a \in P(1..n)$ or
$\exists Xb \in A(r)$: $-\Delta^{X^{l_c}} b \in P(1..n)$ or
$\exists \neg Yd \in A(r)$: $+\Delta^{Y^{l_c}} d \in P(1..n)$

Definite Provability for Prohibitions and Permissions Definite proof conditions for prohibitions can be simply obtained from the ones for O.

$$\pm\Delta^{\mathrm{F}^{l_c}}: \text{ If } P(n+1) = \pm\Delta^{\mathrm{F}^{l_c}} q, \text{ then } \pm\Delta^{\mathrm{O}^{l_c}} {\sim}q \in P(1..n).$$

The concept of permission is much more elusive (for a discussion, see, e.g., [17]). Here, we minimize complexities by adopting perhaps the simplest option among those discussed in [11]. Such an option models permissive norms with defeaters for obligations: a defeater like $a_1, \ldots, a_n \leadsto^{l}_{\mathrm{O}} q$ states that some q is permitted (Pq) in the license l, since it is meant to block deontic defeasible rules for $\sim q$, i.e., rules supporting O$\sim q$[24]. This reading suggests that permissions are only defeasible, hence we postpone the proof theory for permission to the section dealing with the non-monotonic part of the theory[25].

Defeasible Provability As usual in standard Defeasible Logic, to show that a literal q is defeasibly provable we have two choices: (1) we show that q is already definitely provable; or (2) we need to argue using the defeasible part of a multi-license theory D. For this second case, some (sub)conditions must be satisfied. First, we need to consider possible reasoning chains in support of $\sim q$ with the modes l_c and X^{l^c}, and show that $\sim q$ is not definitely provable with that mode (2.1 below). Second, we require that there must be a strict or defeasible rule with mode at hand for q which can apply (2.2 below). Third, we must consider the set of all rules which are not known to be inapplicable and which permit to get $\sim q$ with the mode under consideration (2.3 below). Essentially, each rule s of this kind attacks the conclusion q. To prove q, s must be counterattacked by a rule t for q with the following properties: i) t must be applicable, and ii) t must prevail over s. Thus each attack on the conclusion q must be counterattacked by a stronger rule. In other words, r and the rules t form a team (for q) that defeats the rules s.

Obligation Defeasible Provability

$+\partial^{\mathrm{O}^{l_i}}$: If $P(n+1) = +\partial^{\mathrm{O}^{l_i}} q$ then
(1)$+\Delta^{\mathrm{O}^{l_i}} q \in P(1..n)$ or
(2) (2.1) $-\Delta^{\mathrm{O}^{l_i}} {\sim}q \in P(1..n)$ and
  (2.2) $\exists r \in R^{\mathrm{O}^{l_i}}_{\mathrm{sd}}[q] : \forall a, Xb, \neg Yd \in A(r)$: $+\partial^{c} a, +\partial^{X^{l_i}} b$, $-\partial^{Y^{l_i}} d \in P(1..n)$ and
  (2.3) $\forall l_j \in \mathrm{Lic}, \forall s \in R^{\mathrm{O}^{l_j}}[{\sim}q]$, either
    (2.3.1) $\exists a \in A(s)$ or $Xb \in A(s)$ or $\neg Y \in A(s)$:
      $-\partial^{c} a \in P(1..n)$, or $-\partial^{X^{l_c}} b \in P(1..n)$, or $+\partial^{Y^{l_c}} d \in P(1..n)$; or
    (2.3.2) $\forall l_k \in \mathrm{Lic}, \exists t \in R^{\mathrm{O}^{l_k}}[q]$: $\forall a, Xb, \neg Yd \in A(t)$,
      $+\partial^{c} a, +\partial^{l_c} b, -\partial^{l_c} d \in P(1..n)$, and $t \succ s$.

---

[24] Hence, we do not make explicit in the language the distinction between the cases where we have explicit permissive clauses for P (strong permissions of q [25]) from those where some q is permitted (Pq) because it can be obtained from the fact that $\neg q$ is not provable as mandatory (weak permission). For an extensive treatment of defeasible permissions, see also [10].

[25] For space reasons, we will omit the proof conditions for $-\partial^{\mathrm{O}^{l_i}}$, and $-\partial^{\mathrm{P}^{l_i}}$, which can all be obtained applying the so-called Principle of Strong Negation [11], as illustrated for $-\Delta^{\mathrm{O}^{l_i}}$.

$+\partial^{\mathrm{P}^{l_i}}$: If $P(n+1) = +\partial^{\mathrm{P}^{l_i}} q$ then
(1) (1.1) $-\Delta^{\mathrm{O}^{l_i}} {\sim}q \in P(1..n)$ and
(1.2) $\exists r \in R^{\mathrm{O}^{l_i}}_{\mathrm{dft}}[q] : \forall a, Xb, \neg Yd \in A(r)$: $+\partial^c a, +\partial^{X^{l_i}} b$, $-\partial^{Y^{l_i}} d \in P(1..n)$ and
(1.3) $\forall l_j \in \mathrm{Lic}$, $\forall s \in R^{\mathrm{O}^{l_j}}[{\sim}q]$, either
(1.3.1) $\exists a \in A(s)$ or $Xb \in A(s)$ or $\neg Y \in A(s)$:
$-\partial^c a \in P(1..n)$, or $-\partial^{X^{l_c}} b \in P(1..n)$, or $+\partial^{Y^{l_c}} d \in P(1..n)$; or
(1.3.2) $\forall l_k \in \mathrm{Lic}$, $\exists t \in R^{\mathrm{O}^{l_k}}_{\mathrm{dft}}[q]$: $\forall a, Xb, \neg Yd \in A(t)$,
$+\partial^c a, +\partial^{l_c} b, -\partial^{l_c} d \in P(1..n)$, and $t \succ s$.

Let us consider two examples that illustrate some aspects of the proof theory, and how the heuristics are used before to formally introduce them.

Example 1. Assume to work with two licenses l_1 and l_2 and their composition, and let us reason only about obligations and permissions:

$$
\begin{aligned}
F &= \{a, d\} \\
R^{\mathrm{O}^{l_1}} &= \{r_1 : a \Rightarrow^{l_1}_{\mathrm{O}} p, \qquad r_2 : d \Rightarrow^{l_1}_{\mathrm{O}} {\sim}e\} \\
R^{\mathrm{O}^{l_2}} &= \{r_3 : \Rightarrow^{l_2}_{\mathrm{O}} {\sim}p, \qquad r_4 : a, d \rightsquigarrow^{l_2}_{\mathrm{O}} p, \qquad r_5 : \mathrm{P}p \Rightarrow^{l_2}_{\mathrm{O}} {\sim}e\} \\
\succ &= \{r_4 \succ r_3\}
\end{aligned}
$$

Let us consider AND-composition heuristics only. Rule r_1 leads in l_1 to $+\partial^{\mathrm{O}^{l_1}} p$ (i.e., $\mathrm{O}p$ in l_1). License l_2 supports $+\partial^{\mathrm{P}^{l_2}} p$ because the defeater r_4 is applicable and is stronger than r_3: hence, AND-composition states that $+\partial^{\mathrm{P}^{l_c}} p$ is the case (i.e., that p is permitted in the composite license). This last conclusion triggers r_5 thus obtaining in l_2 the conclusion $+\partial^{\mathrm{O}^{l_2}} {\sim}e$, the same deontic conclusion that is also obtained in l_1 by successfully applying r_2: hence, $+\partial^{\mathrm{O}^{l_c}} {\sim}e$ (i.e., e is prohibited in l_c).

Example 2. Consider two software libraries associated to licenses l_1 and l_2, respectively. License l_1 permits *Commercial* and obliges for *Attribution*, while license l_2 prohibits *Commercial*, permits *Derivative*, and obliges for *ShareAlike*.

$$
\begin{aligned}
L &= \{l_1, l_2\} \\
R^{\mathrm{O}^{l_1}} &= \{r_1 : \Rightarrow^{l_1}_{\mathrm{O}} Attribution, \qquad r_2 : \rightsquigarrow^{l_1}_{\mathrm{O}} Commercial\} \\
R^{\mathrm{O}^{l_2}} &= \{r_3 : \Rightarrow^{l_2}_{\mathrm{O}} {\sim}Commercial, \quad r_4 : \Rightarrow^{l_2}_{\mathrm{O}} ShareAlike, \qquad r_5 : \rightsquigarrow^{l_2}_{\mathrm{O}} Derivative\}
\end{aligned}
$$

We have to decide which heuristics better suits our needs with respect to the single licenses to compose. If we do not include the obligations present in each single license (*Attribution, ShareAlike*), we are not compliant with their normative semantics thus we violate them. To avoid that, the OR heuristics is used to compose obligations. Concerning permissions (*Derivative, Commercial*), we must check that every single license includes the specific permission, thus we adopt the AND heuristics. Otherwise, if there is a prohibition (${\sim}$*Commercial*), and we include the permission in l_c, we violate it. Hence, $+\partial^{\mathrm{O}^{l_c}} Attribution$, $+\partial^{\mathrm{O}^{l_c}} ShareAlike$, and $+\partial^{\mathrm{P}^{l_c}} Derivative$.

The proof conditions for composite licenses we define in the next section assume appropriate definitions for establishing whenever a deontic effect is entailed in a given license, as we presented.

3.4 Provability for composite licenses

According to the OR-composition and AND-composition heuristics, we may compose the deontic effects when each of them is entailed either by at least one license or by all licenses. This idea is directly captured as follows:

OR-composition For $\# \in \{\Delta, \partial\}$ and $X \in \{\mathrm{O}, \mathrm{P}, \mathrm{F}\}$:
$+\#^{X^{l_c}}$: If $P(n+1) = +\#^{X^{l_c}} p$, then $\exists l_i \in \mathrm{Lic} : +\#^{X^{l_i}} p \in P(1..n)$.
$-\#^{X^{l_c}}$: If $P(n+1) = -\#^{X^{l_c}} p$, then $\forall l_i \in \mathrm{Lic} : -\#^{X^{l_i}} p \in P(1..n)$.

AND-composition For $\# \in \{\Delta, \partial\}$ and $X \in \{\mathrm{O}, \mathrm{F}\}$:
$+\#^{X^{l_c}}$: If $P(n+1) = +\#^{X^{l_c}} p$, then $\forall l_i \in \mathrm{Lic} : +\#^{X^{l_i}} p \in P(1..n)$.
$-\#^{X^{l_c}}$: If $P(n+1) = -\#^{X^{l_c}} p$, then $\exists l_i \in \mathrm{Lic} : -\#^{X^{l_i}} p \in P(1..n)$.
$+\partial^{\mathrm{P}^{l_c}}$: If $P(n+1) = +\partial^{\mathrm{P}^{l_c}} p$, then $\exists l_i \in \mathrm{Lic} : +\partial^{\mathrm{P}^{l_i}} p \in P(1..n)$ and
$\forall l_k \in \mathrm{Lic}, l_i \neq l_k$ either $+\partial^{\mathrm{P}^{l_k}} p \in P(1..n)$ or $-\partial^{\mathrm{O}^{l_k}} {\sim}p \in P(1..n)$.
$-\partial^{\mathrm{P}^{l_c}}$: If $P(n+1) = -\partial^{\mathrm{P}^{l_c}} p$, then $\forall l_i \in \mathrm{Lic}$ either $-\partial^{\mathrm{P}^{l_i}} p$ or $+\partial^{\mathrm{P}^{l_i}} {\sim}p$.

The conditions for obligations and prohibitions directly implement what we have informally said in regard to the two heuristics. A brief comment about permissions in AND-composition: we may establish here that some p is permitted in l_c when it is explicitly permitted (via defeaters) in all licenses, or when there is at least one license explicitly permitting p and, in all the other licenses where no explicit permission for p succeeds, p is at least not prohibited (so p is weakly permitted [25,10]).

3.5 Properties and admissibility

The logic presented here is a variant of the one developed in [8,9]. On account of this fact, two results can be imported here: its soundness and computational complexity.

Theorem 1 *Let D be a multi-license theory where the transitive closure of $\succ$ is acyclic. For every $\# \in \{\Delta, \partial\}$, $X \in \{l, Y^l | l \in \mathrm{Lic}, Y \in \{\mathrm{O}, \mathrm{F}\}\}$, and $Z \in \{l, W^l | l \in \mathrm{Lic}, W \in \mathrm{MOD}\}$:*

- *It is not possible that both $D \vdash +\#^Z p$ and $D \vdash -\#^Z p$;*
- *For all $l \in L \cup \{l_c\}$, it is not possible that both $D \vdash +\partial^{\mathrm{O}^l} p$ and $D \vdash +\partial^{\mathrm{P}^l} {\sim}p$;*
- *If $D \vdash +\partial^X p$ and $D \vdash +\partial^X {\sim}p$, then $D \vdash +\Delta^X p$ and $D \vdash +\Delta^X {\sim}p$.*

Given a multi-license theory D, the *universe* of D (U^D) is the set of all the atoms occurring in D. The *extension* (or conclusions) E^D of D is a structure $(\Delta_D^+, \Delta_D^-, \partial_D^+, \partial_D^-)$, where for $X^l \in \mathrm{MOD}$ and $l \in L$:

$$\Delta_D^{\pm} = \{Xq : D \vdash \pm\Delta^{X^l} q\} \cup \{q : D \vdash \pm\Delta^c q\} \qquad \partial_D^{\pm} = \{Xq : D \vdash \pm\partial^{X^l} q\} \cup \{q : D \vdash \pm\partial^c q\}.$$

Theorem 2 *Let $D = (F, L, R^c, \{R^{\mathrm{O}^l}\}_{l \in \mathrm{Lic}}, \succ)$ be a multi-license theory. The extension of D can be computed in time linear to the size of the theory, i.e., $O(|R^c \cup \{R^{\mathrm{O}^l}\}_{l \in \mathrm{Lic}}| * |U^D| * |L|)$.*

Finally, let us establish when a license composition l_c is meaningful or admissible. This can be checked taking into account the following guidelines:

- When only defeasible rules and defeaters are considered, a composition is admissible *iff* it leads to a non-empty set of deontic conclusions. Defeasible Logic is skeptical logic, so in case there is no way to solve deontic conflicts (according to any given heuristics), it means that the composite license does not produce any effect.
- In the case of conflicting strict rules there is no way to block contradictory conclusions. Hence, checking if a composition is admissible also requires to exclude that Δ_D^+ contains contradictory conclusions.
- Facts are supposed to describe a given situation where licenses are applied, thus they can vary from context to context. Hence, we may have two levels for detecting unsolvable conflicts in the licenses' composition: when we consider specific sets of facts, or when we examine licenses in general.

The following definition formally considers all these aspects:

Definition 2. *Let $D = (F, L, \{R^l\}_{l \in \mathrm{Lic}}, \{R^{\mathrm{O}^l}\}_{l \in \mathrm{Lic}}, \{R^{\mathrm{P}^l}\}_{l \in \mathrm{Lic}}, \succ)$ be a multi-license theory and $\mathscr{A}_D$ is the set of all literals and modal literals occurring in the antecedent of all rules of D. The license $l_c = l_1 \odot \cdots \odot l_n$ is F*-admissible *iff*

- *$L = \{l_1, \ldots, l_n\}$,*
- *$\exists X^{l_c} q \in \partial_D^+$, and*
- *for any literal p, if $X \in \{l, Y^l | l \in \mathrm{Lic}, Y \in \{\mathrm{O}, \mathrm{F}\}\}$, then we do not have that $D \vdash +\Delta^X p$ and $D \vdash +\Delta^X {\sim} p$.*

The composite license $l_c = l_1 \odot \cdots \odot l_n$ is admissible *iff it is F-admissible for all $F \subseteq \mathscr{A}_D$.*

4 Mapping into SPINdle and results

In this section we illustrate how to implement the logic and the heuristics developed in Section 3 in SPINdle[26]. SPINdle [15] is a modular and efficient reasoning engine, written in Java, for defeasible logic and modal defeasible logic implementing and extending the algorithms of [8,9]. It has been experimentally tested against the benchmark of [16] showing that it is able to handle very large theories, i.e., theories with hundredth of thousand rules, indeed the largest theory it has been tested with has 1 million rules.

While SPINdle supports multi-modal defeasible logics, currently it does not support natively the AND and OR heuristics presented in this paper. Therefore, we first have to provide polynomial time transformations to implement the two heuristics.

Definition 3. *Let # be one of the proof tags. Two multi-license theories D_1 and D_2 are equivalent (written $D_1 \equiv D_2$) iff $\forall p, D_1 \vdash \#p$ iff $D_2 \vdash \#p$, i.e., they have the same consequences. Similarly $D_1 \equiv_\Sigma D_2$ means that D_1 and D_2 have the same consequences in the language Σ.*

Definition 4. *A* transformation *is a mapping from multi-license theories to multi-license theories. A transformation T is* correct *iff for all theories D_i, $D \equiv_\Sigma T(D)$ where Σ is the language of D.*

[26] `http://spin.nicta.org.au/spindle/index.html`

The OR-heuristic is implemented by the following transformation[27]:

$$tor(r) = \begin{cases} r\colon A(r) \hookrightarrow p & r \in R^c \\ r\colon A(r) \rightarrow_{O^c} p & r \in R_s^{O^{l_i}},\ l_i \in \text{Lic} \\ r\colon A(r) \Rightarrow_{O^c} p & r \in R_d^{O^{l_i}},\ l_i \in \text{Lic} \\ r\colon A(r) \Rightarrow_{-O^c} {\sim}p & r \in R_{dft}^{O^{l_i}},\ l_i \in \text{Lic} \end{cases}$$

It is immediate to see that tor is a one-to-one transformation. For obligation operators tor flattens all of them into O^c. In fact, for the OR-heuristics we need to prove p with $+\partial^{O^{l_i}}$ for a single license, and thus the set of rules to be considered for clause (2.2) is the set of all strict and defeasible obligation rules for p. For permission we use a particular feature of SPINdle, namely 'negative' modalities. A negative modality (e.g. $-O^c$) behaves on one side as any other modality, but it is in a symmetric conflict with the corresponding positive one (i.e., O^c). Thus it can be used to disprove a conclusion for the positive counterpart without proving it. Hence, it behaves essentially like a defeater. Theorem 1 shows that the transformation of tor into the logic of SPINdle is correct.

Theorem 1. *Let $D = (F, L, R, \succ)$ be a multi-license theory. Let $T(D) = (F, L, \{tor(r) : r \in R\}, \succ)$. Then $D \equiv_\Sigma T(D)$.*

We show now by means of an example how to apply our logic to compose three Web of Data popular licenses using the SPINdle transformation.

Example 3. Assume the data returned by different datasets are associated to the Open Government License[28], the Open Database License[29], and the Attribution-NonCommercial-NoDerivs 2.0 Generic License[30]. The three licenses in a machine-readable format are visualized in Figure 2. The multi-license theory D is as follows:

$$\begin{aligned}
F &= \{Open\} \\
L &= \{l_{OGL}, l_{ODbL}, l_{BY-NC-ND}\} \\
R^{O^{l_{OGL}}} &= \{r_1 :\Rightarrow_O^{l_{OGL}} Attribution, \quad r_2 : Open \rightsquigarrow_O^{l_{OGL}} Publishing, \\
&\quad\ r_3 : Open \rightsquigarrow_O^{l_{OGL}} Distribution, \quad r_4 : Open \rightsquigarrow_O^{l_{OGL}} Derivative, \\
&\quad\ r_5 : Open \rightsquigarrow_O^{l_{OGL}} Commercial\} \\
R^{O^{l_{ODbL}}} &= \{r_6 :\Rightarrow_O^{l_{ODbL}} ShareAlike, \quad r_7 :\Rightarrow_O^{l_{ODbL}} Attribution, \\
&\quad\ r_8 :\rightsquigarrow_O^{l_{ODbL}} Sharing, \quad r_9 :\rightsquigarrow_O^{l_{ODbL}} Derivative\} \\
R^{O^{l_{BY-NC-ND}}} &= \{r_{10} :\Rightarrow_O^{l_{BY-NC-ND}} Attribution, \quad r_{11} :\Rightarrow_O^{l_{BY-NC-ND}} {\sim}Commercial, \\
&\quad\ r_{12} :\Rightarrow_O^{l_{BY-NC-ND}} {\sim}Derivative, \quad r_{13} :\rightsquigarrow_O^{l_{BY-NC-ND}} Sharing\} \\
\succ &= \{l_{ODbL} \succ l_{BY-NC-ND}\}
\end{aligned}$$

[27] In the remainder, O^c and P^c abbreviate O^{l_c} and P^{l_c}.

[28] http://www.nationalarchives.gov.uk/doc/open-government-licence/

[29] http://opendatacommons.org/licenses/odbl/

[30] http://creativecommons.org/licenses/by-nc-nd/3.0/

We have now to build the composite license such that $l_c = l_{OGL} \odot l_{ODbL} \odot l_{BY-NC-ND}$. AND-composition is admissible since there is at least one deontic effect entailed by all licenses, i.e., from rules r_1, r_7 and r_{10}, which lead to the deontic conclusion $+\partial^{\mathrm{O}^{l_c}} Attribution$. OR-composition is admissible too: notice that a conflict arises between rule r_5 and rule r_{11} and between rule r_{12} and rules r_4 and r_9. The deontic conclusions are: $+\partial^{\mathrm{O}^{l_c}} Attribution$, $+\partial^{\mathrm{O}^{l_c}} ShareAlike$, $+\partial^{\mathrm{P}^{l_c}} Publishing$, $+\partial^{\mathrm{P}^{l_c}} Distribution$, $+\partial^{\mathrm{P}^{l_c}} Sharing$, $-\partial^{\mathrm{P}^{l_c}} Derivative$, $-\partial^{\mathrm{P}^{l_c}} Commercial$. The tor transformation is

```
>> Open
r1: =>[Oc]Attribution
r2: Open =>[-Oc] -Publishing
r3: Open =>[-Oc] -Distribution
r4: Open =>[-Oc] -Derivative
r5: Open =>[-Oc] -CommercialExpl

r6: =>[Oc] ShareAlike
r7: =>[Oc] Attribution
```

```
r8: =>[-Oc] -Share
r9: =>[-Oc] -Derivative

r10: =>[Oc] Attribution
r11: =>[Oc] -CommercialExpl
r12: =>[Oc] -Derivative
r13: =>[-Oc] -Share

r9 > r12
```

When the above theory is loaded in SPINdle it takes 14 milliseconds to produce the following conclusions `+d [Oc]Attribution`, `+d [-Oc]-Distribution`, `+d [-Oc]-Publishing`, `+d [-Oc]-Share`, `+d [Oc]ShareAlike`, where `+d [Oc]` corresponds to $+\partial^{\mathrm{O}^c}$, and `+d [-Oc]-` means $+\partial^{\mathrm{P}^c}$. Figure 2.d shows the machine-readable l_c.

```
@prefix l4lod: http://ns.inria.fr/l4lod/.
@prefix : http://example/licenses.

:licOGL a l4lod:License;
       l4lod:licensingTerms <http://www.nationalarchives.gov.uk/
                                   doc/open-government-licence/>;
       l4lod:permits l4lod:Publishing;
       l4lod:permits l4lod:Distribution;
       l4lod:permits l4lod:Derivative;
       l4lod:permits l4lod:CommercialExpl;
       l4lod:obliges l4lod:Attribution.
```

(a)

```
@prefix l4lod: http://ns.inria.fr/l4lod/.
@prefix : http://example/licenses.

:licODbL a l4lod:License;
         l4lod:licensingTerms <http://opendatacommons.org/
                                            licenses/odbl/>;
         l4lod:permits l4lod:Sharing;
         l4lod:permits l4lod:Derivative;
         l4lod:obliges l4lod:Attribution;
         l4lod:obliges l4lod:ShareAlike.
```

(b)

```
@prefix cc: http://creativecommons.org/ns.
@prefix l4lod: http://ns.inria.fr/l4lod/.
@prefix : http://example/licenses.

:licBY-CC-NC-ND a cc:License;
         cc:legalcode <http://creativecommons.org/licenses
                                            /by-nc-nd/>;
         cc:permits cc:Sharing;
         cc:requires cc:Attribution;
         cc:prohibits cc:CommercialUse;
         l4lod:prohibits l4lod:NoDerivative.
```

(c)

```
@prefix l4lod: http://ns.inria.fr/
l4lod/.
@prefix : http://example/licenses.

:licComposite a l4lod:License;
   l4lod:obliges l4lod:Attribution;
   l4lod:obliges l4lod:ShareAlike;
   l4lod:permits l4lod:Publishing;
   l4lod:permits l4lod:Distribution;
   l4lod:permits l4lod:Sharing.}
```

(d)

Figure 2. Licenses to be composed (a-b-c) and the resulting composite license (d).

Notice that the overhead introduced by the licenses composition framework is constituted by the query execution time to retrieve the licenses associated to the triples returned as query result (if the set of licenses is known, it can be pre-computed), plus the SPINdle overhead in computing the composite license. For these reasons, we can say that the actual overhead is represented by SPINdle but, as shown above, it does not have a serious impact on query execution time (few milliseconds).

For the AND-heuristic, for a multi-license theory $D = (F, L = \{l_1, \dots, l_n\}, R, \succ)$, the transformation is based on the following sets of rules:

$$tand(r) = \{r_{ij}\colon A(r) \rightsquigarrow_{\mathrm{O}^j} C(r) | r \in R^{\mathrm{O}^{l_i}}\} \cup \{r | r \in R^{\mathrm{O}^{l_i}}_{\mathrm{sd}}\} \cup \{r\colon A(r) \Rightarrow_{-\mathrm{O}^i} C(r) | r \in R^{\mathrm{O}^{l_i}}_{\mathrm{dft}}\}$$

$$R^* = \{o_q\colon \mathrm{O}^1 q, \dots, \mathrm{O}^n q \Rightarrow_{\mathrm{O}^c} q, \quad p_q\colon \mathrm{P}^* q, \mathrm{P}^1 q, \dots, \mathrm{P}^n q \Rightarrow_{\mathrm{P}^c} q$$

$$p_q^{i*}\colon -\mathrm{O}^i{\sim}q \Rightarrow_{\mathrm{P}^*} q, \quad p_q^i\colon -\mathrm{O}^i{\sim}q \Rightarrow_{\mathrm{P}^i} q, \quad f_q^i\colon \neg\mathrm{O}^i q \Rightarrow_{\mathrm{P}^i} q \mid$$

$$l_i \in L, \exists r \in R^{\mathrm{O}^{l_i}}, C(r) = q\}$$

Defeaters are modeled as in the *tor* transformation. The intuition behind the rules r_{ij} is that all rules can be use to attack any other rule, irrespectively of the license. Thus for every obligation rule in a license we create a defeater with the same content for each other license. For the AND-composition of obligations we need that a literal q is provable as obligation in every license; this is achieved by rule o_q. We have to do the same for permissions. However, permission requires that at least one license permits q and for all other licenses q is either permitted or not forbidden. To achieve this we create a 'special' modality P^* and rules linking this to provability of the negative modality $-\mathrm{O}^i$ (encoding permission in SPINdle). Finally, for each license we create its permission modality P^i and a literal q can be derived with such modality if it is permitted in license l_i ($-\mathrm{O}^i q$, rules p_q^i), or if q is not forbidden by that license (i.e., $\neg\mathrm{O}^i q$, rules f_q^i).

Theorem 2 shows that the transformation of *tand* into the logic of SPINdle is correct.

Theorem 2. *Let* $D = (F, L, R, >)$ *be a multi-license theory.* $T(D) = (F, L, \{r | r \in R^c\} \cup \{tand(r) | r \in R^{\mathrm{O}^{l_i}}\}_{l_i \in \mathrm{Lic}} \cup R^*, \succ \cup \{(r_{ij}, s_{ij}), (r_{ij}, s), (r, s_{ij}) | r \succ s\}_{l_i, l_j \in \mathrm{Lic}})$. *Then* $D \equiv_\Sigma T(D)$.

5 Related work

The logic we presented is an extension of the logic of [8,9]. The debt on previous work is the general idea of the formalism, and the proof theory for obligations. What is new is the way for composing licenses (Section 3.4), and for computing permissions and relative results (Section 3.5). Also, we proved that there is a transformation mapping the new logic into SPINdle.

Pucella and Weissman [20] propose a logic to check whether the user's actions follow the licenses' specifications. They do not deal with composition and do not provide a deontic account of licenses' conclusions. Furthermore, their logic is not able to handle conflicting licenses.

Nadah et al. [19] propose to assist licensors' work by providing them a generic way to instantiate licenses, independent from specific formats. We go towards the definition of a composite license while they go towards the definition of a specific ontology (about 100 concepts) used for the translation in the different formats.

Gangadharan et al. [6] address the issue of service license composition and compatibility analysis basing on ODRL-S, an extension of ODRL to implement the clauses of service licensing. They specify a matchmaking algorithm which verifies whether two service licenses are compatible. In case of a positive answer, the services can be composed and the framework determines the license of the composite service. Truong et al. [22] address the issue of analyzing data contracts, based on ODRL-S. Contract analysis leads to the definition of a contract composition where first the comparable contractual terms from the different data contracts are retrieved, and second an evaluation of the new contractual terms for the data mash-up is addressed. Villata and Gandon [23] follow a similar approach to evaluate the compatibility of CC licenses and compose them. There are several differences w.r.t. these approaches: *(i)* the application scenario is different (service composition vs. Web of Data); *(ii)* we allow for a normative reasoning which goes beyond basic compatibility rules by exploiting normative compliance. However, common points are the idea of merging the clauses of the different licenses/contracts, and the use of RDF for licenses/contracts representation.

Krotzsch and Speiser [14] present a semantic framework for evaluating ShareAlike recursive statements. In particular, they develop a general policy modelling language, then instantiated with OWL DL and Datalog, for supporting self-referential policies as expressed by CC. In this paper, we address another kind of problem that is the composition of the deontic components of single licenses into a composite license.

Gordon [7] presents a legal prototype for analyzing open source licenses compatibility using the Carneades argumentation system. Licenses compatibility is addressed at a different granularity w.r.t. our purpose, and licenses composition is not considered.

The attachment of additional information like rights or licenses to RDF triplets is linked to an active research field. Carroll et al. [5] introduced Named Graphs in RDF to allow publishers to communicate assertional intent and to sign their assertions. Moreover, the W3C Provenance WG defines the kind of information to be used to form assessments about data quality, reliability or trustworthiness [12].

6 Conclusions

In this paper, we propose an automated framework for licenses composition based on deontic logic. The rationale behind this framework is to build a composite license starting from the single licensing terms associated to heterogeneous data. We adopt deontic logic to ensure the compliance of the composite license with respect to the single licenses composing it. We evaluate the feasibility of the automatic generation of the composite license on the SPINdle defeasible reasoner.

There are several lines to pursue as future research. First, we will develop a standalone licensing module generating the machine-readable composite license every time a query returns multi-licensed data. Second, we still have to consider the case of data obtained by inference from one or several licensed datasets. In particular, a special case we have to address is the one of queries going beyond basic `SELECT` queries, where aggregations are present, e.g., return the *average*, *sum*, etc. of the data possibly over distributed datasets. Third, the logic should take into account the temporal aspect of licenses. In particular, two concepts to be considered are *validity time* (point in time where a deontic component is true) and *reference time* (point in time the obligation, prohibition or permission applies to) of an obligation, prohibition or permission. Finally, even if our framework allows to reason about certain characteristics of licenses, e.g., whether attribution is required or commercial usages are permitted, it is still an open problem the fact that there is no uniform, cross-national definition of essential legal terms. We will investigate suitable solutions with further legal experts.

References

1. Abelson, H., Adida, B., Linksvayer, M., Yergler, N.: ccREL: The creative commons rights expression language. Tech. rep. (2008)
2. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation results for defeasible logic. ACM Transactions on Computational Logic 2(2), 255–287 (2001)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)
4. Boella, G., Governatori, G., Rotolo, A., van der Torre, L.: A logical understanding of legal interpretation. In: Proceedings of KR. AAAI Press (2010)
5. Carroll, J., Bizer, C., Hayes, P., Stickler, P.: Named graphs. J. Web Sem. 3(4), 247–267 (2005)
6. Gangadharan, G.R., Weiss, M., D'Andrea, V., Iannella, R.: Service license composition and compatibility analysis. In: Proceedings of ICSOC, LNCS 4749. pp. 257–269. Springer (2007)
7. Gordon, T.F.: Analyzing open source license compatibility issues with Carneades. In: Proceedings of ICAIL. pp. 51–55. ACM (2011)
8. Governatori, G., Rotolo, A.: BIO logical agents: Norms, beliefs, intentions in defeasible logic. Autonomous Agents and Multi-Agent Systems 17(1), 36–69 (2008)
9. Governatori, G., Rotolo, A.: A computational framework for institutional agency. Artif. Intell. Law 16(1), 25–52 (2008)
10. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Three concepts of defeasible permission. In: Proceedings of JURIX. pp. 63–72. IOS Press (2011)
11. Governatori, G., Padmanabhan, V., Rotolo, A., Sattar, A.: A defeasible logic for modelling policy-based intentions and motivational attitudes. Logic J. of IGPL 17(3), 227–265 (2009)
12. Groth, P.T., Gil, Y., Cheney, J., Miles, S.: Requirements for provenance on the web. IJDC 7(1), 39–56 (2012)
13. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool (2011)
14. Krötzsch, M., Speiser, S.: ShareAlike Your Data: Self-referential Usage Policies for the Semantic Web. In: Proceedings of ISWC, LNCS 7031. pp. 354–369. Springer (2011)
15. Lam, H.P., Governatori, G.: The making of SPINdle. In: Proceedings of RuleML, LNCS 5858. pp. 315–322. Springer (2009)
16. Maher, M.J., Rock, A., Antoniou, G., Billington, D., Miller, T.: Efficient defeasible reasoning systems. International Journal of Artificial Intelligence Tools 10, 483–501 (2001)
17. Makinson, D., van der Torre, L.: Permission from an input/output perspective. Journal of Philosophical Logic 32(4), 391–416 (2003)
18. Miller, P., Styles, R., Heath, T.: Open data commons, a license for open data. In: Proceedings of LDOW (2008)
19. Nadah, N., de Rosnay, M.D., Bachimont, B.: Licensing digital content with a generic ontology: escaping from the jungle of rights expression languages. In: Proceedings of ICAIL. pp. 65–69. ACM (2007)
20. Pucella, R., Weissman, V.: A logic for reasoning about digital rights. In: Proceedings of CSFW. pp. 282–294. IEEE (2002)
21. Rotolo, A., Villata, S., Gandon, F.: A deontic logic semantics for licenses composition in the web of data. In: Proceedings of ICAIL. pp. 111–120. ACM (2013)
22. Truong, H.L., Gangadharan, G.R., Comerio, M., Dustdar, S., Paoli, F.D.: On analyzing and developing data contracts in cloud-based data marketplaces. In: Proceedings of APSCC, IEEE. pp. 174–181 (2011)
23. Villata, S., Gandon, F.: Licenses compatibility and composition in the web of data. In: Proceedings of COLD. CEUR Workshop Proceedings 905 (2012)
24. von Wright, G.: Deontic logic. Mind 60(237), 1–15 (1951)
25. von Wright, G.: Norm and action: A logical inquiry. Routledge and Kegan Paul (1963)