Heuristics for Licenses Composition

Guido GOVERNATORI [a,1], Ho-Pun LAM [a], Antonino ROTOLO [b], Serena VILLATA [c] and Fabien GANDON [c]
[a] *NICTA Queensland Research Laboratory*
[b] *University of Bologna*
[c] *INRIA Sophia Antipolis*

Abstract. The Web of Data is assisting to a growth of interest with respect to the open challenge of representing and reasoning in an automated way over licenses and copyright. In this paper, we deal with the problem of checking the composing together a set of licensing terms associated to a single query result on the Web of Data to create a so called *composite* license. More precisely, we analyze two composition heuristics, *AND-composition* and *OR-composition*, showing how they can be used to combine the deontic components specified by the licenses, i.e., permissions, obligations, and prohibitions, and which are the most suitable combinations depending on the starting licenses. Such heuristics are evaluated using the SPINdle logic reasoner.

Introduction

In the Web of Data [11], the problem of handling the licensing terms associated to the data in an automated way is becoming more and more important. Several challenges arise to represent licensing information in a machine-readable format (i.e., from the definition of lightweight vocabularies like ORDL[2] and Creative Commons[3] up to the definition of specific ontology design patterns), and to reason over such information to achieve more complex goals like checking the compatibility of a set of licenses and compose them in a compliant way. In particular, the problem of combining the set of terms belonging to heterogeneous licenses or contracts has been studied in different contexts [4,3,18,17]. However, a deeper analysis of the licenses composition heuristics w.r.t. their deontic component (i.e., *permissions*, *obligations*, and *prohibitions*) is needed.

In this paper, we address the research question: how to reason over the composition of a set of licenses such that their deontic component guides the choice of the heuristic? To answer this question we rely on the defeasible deontic logic presented in [17,10], and we include two composition heuristics, namely *AND-composition* and *OR-composition*, in the SPINdle reasoning engine [13] to evaluate their computational feasibility

Figure 1 shows the overall workflow of the licenses composition framework. Our application scenario consists in a data consumer querying a SPARQL endpoint to obtain some data (step 1, Fig. 1). We retrieve the (possibly numerous) license(s) associated to the triples of query result (steps 1-2, Fig. 1), and if more than one license is associated to such

[1]Corresponding Author: `guido.governatori@nicta.com.au`
[2]`http://w3.org/ns/odrl/2/`
[3]`http://creativecommons.org/ns`

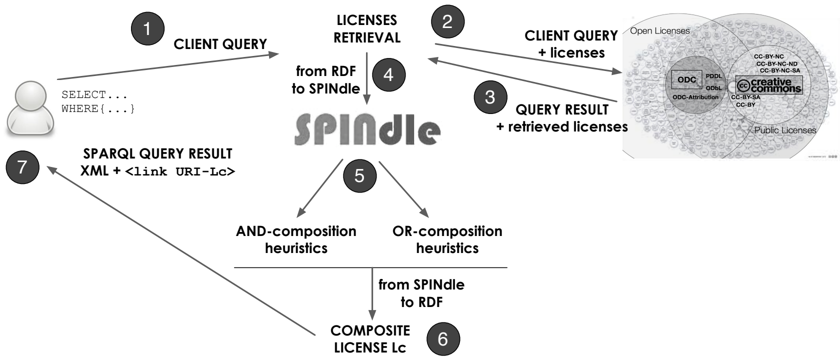

Figure 1. Workflow of our licenses composition framework.

triples, we compose them into a unique *composite license*. We first translate the retrieved licenses from RDF to the SPINdle syntax (step 4, Fig. 1) and then the whole theory, containing all the licenses to be composed, is loaded in SPINDdle. SPINdle manages two kinds of composition heuristics (step 5, Fig. 1): the AND-composition (i.e., the composite license entails a deontic effect if all the licenses composing it entail such deontic effect), and the OR-composition (i.e., the composite license entails a deontic effect if there is at least one licenses that entails such effect, and no license prevents it). These two heuristics can be combined together to produce the composite licenses (step 6, Fig. 1), allowing in such a way a different treatment for each deontic component. Finally, we return to the consumer the query result together with the URI of the machine-readable composite license (step 7, Fig. 1).

The limitations of our model are: *(i)* we do not consider other composition heuristics like the *Constraining Value* and quantitative heuristics [3], and *(ii)* our composition framework does not consider the *additional terms* of the licenses. Finally, note that our application scenario does not deal with *dual-licensing* (alternative licenses for the same data), but we deal with the composition of different licenses associated to different triples which are returned together as result of a SPARQL query.

The remainder of the paper is as follows: Section 1 discusses the related literature, and in Section 2 we formally define the two composition heuristics and evaluate them using the SPINdle reasoner.

1. Related Work

The closest set of related work is in the area of contracts compatibility and composition for services composition. Comerio [3] analyses which kind of qualitative and quantitative heuristics can be used for contacts composition in the context of services composition. Qualitative heuristics include AND- and OR-composition heuristics plus the *Constraining Value* one where the most constraining value among the ones offered by the contracts of the services to be composed is included in the composite service. Quantitative heuristics, instead, include `MIN`, `MAX`, `AVG` and `SUM` (the composite contract offers the minimum (resp. maximum, mean, sum) among the values offered by the contracts of the single services involved in the composition. In this paper, we do not consider quantitative heuristics, and we propose a fine grained analysis of the ADN- and OR-composition heuristics and how to combine them with respect to the deontic component of the single licenses to compose. Gangadharan et al. [4] address the issue of service license composition and compatibility analysis, specifying 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. [18] address a similar problem concerning data contracts: in contracts composition, first the comparable contractual terms from the different contracts are retrieved, and second an evaluation of the new contractual terms for the data mash-up is addressed. We concentrated on the evaluation of the composition heuristics and how they can be composed to better suit the data publisher's needs. Other related work concerns reasoning about licenses [16,5] or licensing issues in the Semantic Web scenario [15,12]. However, they do not address the issue of licenses heuristics composition which is the goal of this paper.

2. Composition heuristics for data licensing

We begin by introducing the defeasible deontic logic we rely on to automatically generate the composite licenses ensuring its compliance w.r.t. the single licenses composing it.

We propose an extension of Defeasible Logic, extending earlier works [9,8], to handle license composition. Previous versions of this logic were proposed in [17,10]. The current version, as will see, is much more compact than the one in [17], and proposes a new and more intuitive reading of AND-composition and OR-composition than the one in [10]. Both improvements allow us to easily generate composite licenses. Moreover, in this paper we present an evaluation of such composition heuristics which is absent in [17], and we implemented the licenses composer into the SPINdle reasoner, differently from [10] where the evaluation resulted from a transformation only. Dealing with license composition 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.

We 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 [2,17]. Notice that we assume that all licenses share a same ontology, or that the ontologies are aligned.

The formal language of the logic is rule-based. Literals can be plain, such as $p, q, r \dots$, 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 $a_1,\dots,a_n \Rightarrow b$ supports the conclusion of b, given $a_1,\dots,a_n$, and so it states that, from the viewpoint of any license any instance enjoying $a_1,\dots,a_n$ is also an instance of b. On the contrary, rules as $a, \mathrm{O}b \Rightarrow^{l_2}_{\mathrm{O}} p$ state that, if a is the case and b is obligatory, then $\mathrm{O}p$ 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 $l_1,\dots,l_n$, the conclusions derived in the logic for $l_1,\dots,l_n$ are also used to establish those that hold in l_c.

Formal language and basic concepts The basic language is defined as follows. Let $\mathrm{Lic} = \{l_1, l_2, \dots, l_n\}$ 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.

Every rule is of the type $r : A(r) \hookrightarrow^{x}_{Y} C(r)$, where: r is a unique identifier for the rule; $A(r) = \{a_1,\dots,a_n\}$, the *antecedent* is a set literal if r is an ontology rule, and a set of modal literals and literals if r is a deontic rule; $C(r)$ the *consequent* is a literal; if r is a deontic rule $Y \in \mathrm{MOD}$ represents the type of conclusion obtained (We will see why we do not need rules for prohibitions and permissions) and $x \in \mathrm{Lic}$ indicates to which license the rule refers to; Y and x are not used for ontology rules.

The intuition behind the different arrows is the following. *Strict rules* have the form $a_1,\dots,a_n \rightarrow^{x}_{Y} b$. *Defeasible rules* have the form $a_1,\dots,a_n \Rightarrow^{x}_{Y} b$. A rule of the form $a_1,\dots,a_n \rightsquigarrow^{x}_{Y} b$ 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* $R^{\mathrm{O}^l}, R^{\mathrm{O}^{l'}} \in \{R^{\mathrm{O}^l}\}_{l\in\mathrm{Lic}}$.

$R[b]$ 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.

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 $X \in \{c, Y^l | l \in \text{Lic}, Y \in \text{MOD}\}$. 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[4].

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.

We concentrate here 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). In [17,10], OR- and AND-compositions were basically characterized as follows:

- **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 this paper, we adopt another approach and associate the different heuristics to the derivation of different deontic effects. In particular, OR-composition allows to establish what *obligations* hold in the composite license: something is obligatory if there is at least one license supporting it. For permissions, instead, AND-composition is the case, as to prove that something is permitted in the composite license we have either to prove that this is the case in all licenses or to exclude that the opposite is obligatory in some license.

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 [1]. 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 each license l_i interplay recursively: indeed, we may use a conclusion for l_c to fire a rule in l_i.

OR-composition and Obligations Let us first define the condition for monotonic derivations of the obligations in each license l_i and the condition for the monotonic derivations of the obligations in the composite license l_c: this second case is a first illustration of the OR-composition heuristics. Assume $x \in \{c, i\}$:[5]

---

[4] 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 $Y = \text{P}$.

[5] We omit the other proof conditions for the deontic effects in each license, defeasible conditions for prohibitions, and all the negative proof conditions, i.e., for $-\Delta^{\text{O}^{l_i}}$, $-\Delta^{\text{O}^{l_c}}$, $-\Delta^{\text{F}^{l_i}}$, $-\Delta^{\text{F}^{l_c}}$, $-\partial^{\text{O}^{l_i}}$, $-\partial^{\text{O}^{l_c}}$, $-\partial^{\text{P}^{l_i}}$, and

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

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

Definite proof conditions for prohibitions can be simply obtained:

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

A second illustration of the OR-composition is offered in the defeasible derivations of the obligations in l_c:

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

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.

AND-composition and Permissions The concept of permission is much more elusive (for a discussion, see, e.g., [14]). Here, we minimize complexities by adopting perhaps the two simplest options among those discussed in [7]. Such options model permissions either as obtained

1. when it is possible to show that the opposite obligations are not provable; or
2. from 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${\sim}q$.

The first type of permissions corresponds to the so-called weak permissions, according to which some q is permitted (Pq) because it can be obtained from the fact that $\neg q$ is not

$-\partial^{\mathrm{P}^{l_c}}$. The negative conditions can be obtained from positive conditions applying the so-called Principle of Strong Negation [7].

provable as mandatory [19]. The second type of permissions is just one way for modeling explicit permissive clauses for proving Pq (strong permissions of q): for an extensive treatment of defeasible permissions, see [6]. This reading suggests that permissions are essentially defeasible.

Permission, version I (Weak Permission)

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

The first type of permission might be useful for combination for 'public domain' type of license, meaning, that unless explicitly obliged or forbidden data can be used freely.

Permission, version II (Strong Permission)

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

The logic presented here is a variant of the one developed in [9,8]. Accordingly, results of soundness and linear computational complexity can be directly imported here [17,10].

The following example illustrate some aspects of the proof theory, and how the heuristics are used.

Example 2 *Consider two datasets published on the LOD cloud*[6] *associated to licenses l_1 and l_2, respectively. License l_1 permits Derivative and obliges for Share-Alike, while license l_2 prohibits Derivative, permits Reproduction, and obliges for Notice.*

$$R^{\mathrm{O}^{l_1}} = \{r_1 :\Rightarrow^{l_1}_{\mathrm{O}} \textit{Share-Alike}, \quad r_2 :\rightsquigarrow^{l_1}_{\mathrm{O}} \textit{Derivative}\}$$
$$R^{\mathrm{O}^{l_2}} = \{r_3 :\Rightarrow^{l_2}_{\mathrm{O}} {\sim}\textit{Derivative}, \quad r_4 :\Rightarrow^{l_2}_{\mathrm{O}} \textit{Notice}, \quad r_5 :\rightsquigarrow^{l_2}_{\mathrm{O}} \textit{Reproduction}\}$$

The data publisher has to decide which heuristics better suits her own needs such that the composite license protects as desired the reuse of the released data. First, she needs to include the obligations present in each single license (Share-Alike, Notice) to be compliant with their normative semantics. Thus OR-composition is used to compose obligations. Concerning permissions (Derivative, Reproduction), she has to check that every single license includes the specific permission, thus adopting AND-composition. Given that license l_2 obliges for ${\sim}$Derivative (i.e., prohibits Derivative), we cannot include such permission in the composite license. Hence, $+\partial^{\mathrm{O}^{l_c}}$ Share-Alike, $+\partial^{\mathrm{O}^{l_c}}$ Notice, and $+\partial^{\mathrm{P}^{l_c}}$ Reproduction.

Heuristics' implementation and evaluation We show now how the heuristics we propose can be used to check the compatibility and combine four real world licenses, widely adopted in the Linked Open Data scenario, using the SPINdle reasoner. The licenses we consider are the Creative Commons Public Domain Mark 1.0[7], the OS Open Data li-

[6] `http://lod-cloud.net/`
[7] `http://creativecommons.org/publicdomain/mark/1.0/`

cense[8], the Creative Commons Attribution-NoDerivs license[9], and the Creative Commons Attribution-NonCommercial-ShareAlike[10]. The deontic component of such licenses is:

- CC PDM
  - Permissions: Reproduction, Distribution, Derivative Works.
- OS OpenData
  - Permissions: Reproduction, Distribution, Derivative Works.
  - Obligations: Notice, Attribution.
- CC-BY-ND
  - Permissions: Reproduction, Distribution.
  - Obligations: Notice, Attribution, Share Alike.
- CC-BY-NC-SA
  - Permissions: Reproduction, Distribution.
  - Obligations: Notice, Attribution.
  - Prohibitions: Commercial.

Notice that licenses CC PDM, OS OpenData and CC-BY-ND allow to make commercial use of the work, i.e., the permission is not explicitly stated but it is ensured by the absence of the obligation for Non Commercial, and that license CC-BY-ND does not permit Derivative Works even if such prohibition is not mentioned.

As mentioned before, we have to verify the compatibility of different licensing terms by composing them into a unique composed theory. The implementation is based on the two transformations proposed in [10] for AND- and OR-compositions.

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

For the OR-heuristic we can use the *tor* transformation as it is. For weak permission SPINdle generates the conclusion $+\partial^{\mathrm{P}^c} q$ as soon as a conclusion $-\partial^{\mathrm{O}^{l_i}} {\sim}q$ is generated (alternatively we could add the following set of rules $\{\neg \mathrm{O}^{l_i} {\sim}q \Rightarrow_{-\mathrm{O}^c} q \mid l_i \in \mathrm{Lic}, q \in \mathrm{Lit}\}$ but this transformation could lead to many additional rules, with a degrade in the performance). For AND-composition, we consider the following transformations:

$$\begin{aligned} tando(r) = {} & \{r_{ij} : A(r) \rightsquigarrow_{\mathrm{O}^j} C(r) | r \in R^{\mathrm{O}^{l_i}}\} \cup \{r : A(r) \Rightarrow_{-\mathrm{O}^i} {\sim}C(r) | r \in R_{dft}^{\mathrm{O}^{l_i}}\} \cup \\ & \{r | r \in R_{sd}^{O^{l_i}}\} \cup \{o_q :\ \mathrm{O}^{l_1} q, \ldots, \mathrm{O}^{l_n} q \Rightarrow_{\mathrm{O}^c} q \mid l_i \in \mathrm{Lic}, \exists r \in R^{\mathrm{O}^{l_i}}, C(r) = q\} \end{aligned}$$

$$\begin{aligned} tandsp(r) = {} & \{p_q :\ \mathrm{P}^* q, \mathrm{P}^{l_1} q, \ldots, \mathrm{P}^{l_n} q \Rightarrow_{\mathrm{P}^c} q, f_q^i :\ \neg \mathrm{O}^{l_i} q \Rightarrow_{P^{l_i}} q \\ & p_q^{i*} :\ -\mathrm{O}^{l_i} {\sim}q \Rightarrow_{P^*} q,\ p_q :\ -\mathrm{O}^{l_i} {\sim}q \Rightarrow_{P^{l_i}} q \mid l_i \in \mathrm{Lic}, \exists r \in R^{\mathrm{O}^{l_i}}, C(r) = q\} \end{aligned}$$

$$tandwp(r) = \{p_q :\ \neg \mathrm{O}^{l_1} {\sim}q, \ldots, \neg \mathrm{O}^{l_n} {\sim}q \Rightarrow_{P^c} q \mid l_i \in \mathrm{Lic}, q \in \mathrm{Lit}\}$$

The transformations *tor*, *tando*, *tandsp* and and *tandwp* are used to map rules in different licenses to the composed theory. The combination of *tando* and *tandsp* gives us the

[8] `http://www.ordnancesurvey.co.uk/docs/licences/os-opendata-licence.pdf`
[9] `http://creativecommons.org/licenses/by-nd/3.0/`
[10] `http://creativecommons.org/licenses/by-nc-sa/3.0/`

transformation given in [10] to handle the AND-composition. However, we can use *tando* and *tandwp* to compute the AND-composition with weak permission instead of strong permission. In addition it is possible to use the *tor* and *tandsp* (or *tandwp*) to model the hybrid composition P-AND/OR-O discussed in Example 2. In fact, for the *OR-heuristic*, to prove $+\partial_{O^c} p$ we only need to prove that $O^c p$ is provable by a license; while in the *AND-heuristic* we have to show that the literal is provable in all licenses. However, the case for permission is a bit different as, in addition to the condition above, it also requires least one license permits p. To this end, we have implemented a *theory composer* to apply the transformations above to different licenses and compose them into a single defeasible theory, before passing it to SPINdle for reasoning.

Even though one may argue that modifying the inference engine or devise a new reasoning algorithm to solve our problem can achieve a better performance. However, our approach can give us, at least, three advantages: (1) the implementation of the licenses composer is a lot simpler when comparing with modifying the reasoning engine or implementing a new inference algorithm; (2) instead of computing the conclusions for permission (*AND-composition*) and obligation (*OR-composition*) with two separate inferences (as described in [10]), the proposed approach enable us to compute all conclusions with one single inference, which is simpler and can reduce the time required for initializing the reasoning engine; (3) we can utilize the features provided by SPINdle to capture different intuitions, i.e., ambiguity propagation, well-founded semantics, or their combinations, in the future without additional work.

Heuristics	Theory compose time (ms)	Reasoning time (ms)	Total time (ms)
NO	6.6	47.8	54.4
OR-composition	11.6	59.6	71.2
P-AND/OR-O-composition	17.3	92.4	109.7
AND-composition	25	210.8	235.8
Both AND- and OR-composition	29	226.2	255.2

Table 1. Reasoning time used for different heuristics

Table 1 shows the composition and reasoning time used for composing the four licenses mentioned in the previous section. As expected, there is not much different in the time required to generate the composed theory (and reasoning) with or without the *OR-composition*, and there is a minimal overhead with the *P-AND/OR-O-composition*. However, the cases with *AND-composition* (the latter two cases) is a bit worse (but still in acceptable range) as plethora of propositions are added to the composed theory to ensure consistency among different licenses.

3. Concluding remarks

The development of new models and tools for the advanced management of legal information and knowledge in the Web of Data raises open challenges. In this paper, we have formally defined and evaluated two heuristics for combining the licensing terms of a set of licenses in a compliant way, using our defeasible deontic logic. The OR- and AND-composition heuristics have been coded into the SPINdle defeasible logic reasoner, and the system's evaluation show the applicability of the proposed formal approach in the Web of Data scenario. Several future directions will be considered. First, we will enlarge the set of composition heuristics taking into account also qualitative ones and the Constraining-value heuristic. Second, we are planning the development of a standalone licensing module able to compose a set of licenses in an automated way, and to generate the machine-readable composite license. Finally, we will investigate the adoption of existing trans-border licensing schemes to address the issue of national legal terms.

References

[1] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. Representation Results for Defeasible Logic. *ACM Transactions on Computational Logic*, 2(2):255–286, 2001.

[2] G. Boella, G. Governatori, A. Rotolo, and L. van der Torre. A Logical Understanding of Legal Interpretation. In *Proceedings of KR'10*. AAAI Press, 2010.

[3] M. Comerio. *Web Service Contracts: Specification, Selection and Composition*. PhD Thesis, University of Milano-Bicocca, Milan, Italy, February 2009.

[4] G. R. Gangadharan, M. Weiss, V. D'Andrea, and R. Iannella. Service License Composition and Compatibility Analysis. In *Proc. ICSOC'07*, pp 257–269. Springer, 2007.

[5] T. F. Gordon. Analyzing open source license compatibility issues with Carneades. In *Proc. ICAIL'11*, pp 51–55. ACM, 2011.

[6] G. Governatori, F. Olivieri, A. Rotolo, and S. Scannapieco. Computing Strong and Weak Permissions in Defeasible Logic. *Journal of Philosophical Logic*, forthcoming.

[7] G. Governatori, V. Padmanabhan, A. Rotolo, and A. Sattar. A defeasible logic for modelling policy-based intentions and motivational attitudes. *Logic Journal of IGPL*, 17(3):227–265, 2009.

[8] G. Governatori and A. Rotolo. A Computational Framework for Institutional Agency. *Artificial Intelligence and Law*, 16(1):25–52, 2008.

[9] G. Governatori and A. Rotolo. BIO Logical Agents: Norms, Beliefs, Intentions in Defeasible Logic. *Journal of Autonomous Agents and Multi Agent Systems*, pages 36–69, 2008.

[10] G. Governatori, A. Rotolo, S. Villata, and F. Gandon. One License to Compose Them All: A Deontic Logic Approach to Data Licensing on the Web of Data. In *Proc ISWC'13*. Springer, 2013.

[11] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool, 2011.

[12] M. Krötzsch and S. Speiser. ShareAlike Your Data: Self-referential Usage Policies for the Semantic Web. In *Proc ISWC'11*, pp. 354–369. Springer, 2011.

[13] H.-P. Lam and G. Governatori. The Making of SPINdle. In A. Paschke, G. Governatori, and J. Hall, editors, *Proc. RuleML'09*, pp. 315–322. Springer-Verlag, 2009.

[14] D. Makinson and L. van der Torre. Permission from an Input/Output Perspective. *Journal of Philosophical Logic*, 32(4):391–416, 2003.

[15] N. Nadah, M. D. de Rosnay, and B. Bachimont. Licensing Digital Content with a Generic Ontology: Escaping From the Jungle of Rights Expression Languages. In *Proc. ICAIL'07*, pp. 65–69. ACM, 2007.

[16] R. Pucella and V. Weissman. A Logic for Reasoning about Digital Rights. In *Proc. CSFW'02*, pp. 282–. IEEE, 2002.

[17] A. Rotolo, S. Villata, and F. Gandon. A Deontic Logic Semantics for Licenses Composition in the Web of Data. In *Proc. ICAIL'13*, pp. . ACM, 2013.

[18] H.-L. Truong, G. R. Gangadharan, M. Comerio, S. Dustdar, and F. De Paoli. On Analyzing and Developing Data Contracts in Cloud-Based Data Marketplaces. In *Proc APSCC 2011*, pp. 174–181, 2011.

[19] G. H. von Wright. *Norm and Action: A Logical Enquiry*. Routledge and Kegan Paul, 1963.