Composing complex licenses to facilitate contextual resources reuse

Manoé Kieffer[1,*], Patricia Serrano-Alvarado[1] and Margo Bernelin[2]

[1]Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France
[2]CNRS, Nantes Université, DCS, UMR 6297, F-44000 Nantes, France

Abstract

The ever-growing number of accessible online resources allows any person to modify, to derive, to combine resources in order to create and share new resources. Overseeing those interactions are licenses that define if and how each resource can be reused. As the author can choose to protect and share the new resource to any extent, the license used can be more or less complex and specific. In particular, contextual constraints in a license can circumscribe the reusability of its covered resource to a specific context such as a given location, during a given timeframe, or in specific media. In addition, the new resource must be protected by a license that complies to the licenses of its source materials, it should also be the least restrictive possible in order to make it as shareable as the author wishes. This makes finding a license which complies to a set of potentially complex licenses a challenging task for an author without legal knowledge. We propose CLiC, an approach that, while taking into account contextual constraints expressed in ODRL, composes the least restrictive license that complies to a given set of potentially complex licenses. The idea is to help authors in reusing resources protected by complex licenses while maximizing the shareability of new resources. This way an author obtains the adequate license for a new resource that complies to the licenses of its source materials, while limiting as little as possible its shareability.

Keywords

Complex licenses, ODRL, License constraints, licenses compatibility, licenses composition, license restrictiveness.

1. Introduction

The amount of available online resources motivates authors to take inspiration from, to modify, to derive, and to compose existing resources into new ones. In that context, attaching a license to a new resource is fundamental to control and facilitate reusability. A license specifies how a resource can and cannot be used, i.e., what actions are permitted, obligated and prohibited. The most well-known licenses used with online resources are simple licenses designed to foster an easier reuse of resources. Their simplicity comes from the use of general and clearly defined clauses, allowing the licenses to be applicable in numerous contexts. Examples of such licenses include the Creative Commons' licenses[1] for creative works, or the MIT[2] and Apache[3] licenses for source code.

However, some cases require resources to be protected by more specific or complex licenses that include contextual constraints. Therefore, in addition to the well-known licenses, a number of other licenses can exist. They, then, vary in terms of their complexity, scope of application, or drafting languages. Online stock image resources, coming from companies such as Shutterstock or Adobe, are an example of resources protected by complex dedicated licenses that describe specific contextual constraints[4][5]. Those constraints limit a resource's reuse to a given context, e.g., a circumscribed timeline, a particular country, specific purpose, a certain number of reuse. Contextual constraints can also include

OPAL 2025: Co-located with the Extended Semantic Web Conference. Portorož, Slovenia. June 1, 2025
*Corresponding author.
manoe.kieffer@univ-nantes.fr (M. Kieffer); patricia.serrano-alvarado@univ-nantes.fr (P. Serrano-Alvarado); margo.bernelin@univ-nantes.fr (M. Bernelin)
0000-0003-2367-5361 (P. Serrano-Alvarado); 0000-0002-7342-3499 (M. Bernelin)

[1]https://creativecommons.org/share-your-work/cclicenses/
[2]https://opensource.org/license/mit
[3]https://www.apache.org/licenses/LICENSE-2.0
[4]https://www.shutterstock.com/license
[5]https://stock.adobe.com/license-terms

specific information such as the legal jurisdiction the licenses must comply with or the court where a case will need to be brought if legal actions were to take place in relation to the covered resource. Contractual by nature, these legal documents should not be overlooked by the resources' users as legal consequences are attached to their infringement both under contract law and intellectual property law.

When combining multiple source materials into a new resource, an author is tasked with finding a license that will protect the new resource and be compliant with the sources' first licenses. The re-user will have the daunting task to delve into the legal analyses of multiple licenses, perhaps in different languages, in order to pick a license that will accommodate the first ones. Indeed, the overall license should, at most, have the same permissions (or a coherent intersection of them), and at least the same obligations and prohibitions than the licenses of the source materials [1]. Those combining principles will also mean that the overall license should be restrictive enough to comply with the licenses of the source materials. Thus, the chosen license should be *compliant* with the licenses of the source materials, while the latter should be *compatible* with the chosen license.

Following the idea of maximizing shareability, the author should also pay attention that his/her license for his/her new resource is not too restrictive as it reduces its reusability. This additional constraint will, again, require the author to legally analyse closely licenses options. If the author does not find a suitable license, he/she will be left to draft his/her own licensing terms. In other words, the author will often be required to draft from scratch a contractual document, the license, that will be legally compliant with the initial licenses covering the source materials. In that scenario, the author will have to compose a new license by combining terms from the original ones without introducing errors or loopholes. This will require the author to have a good legal understanding of the first licenses and the ability to translate his/her analysis into another contractual document. From a legal point of view, any mistake will bare serious and actual contractual consequences. Such legal analysis and drafting are often left to Contract or Intellectual Property lawyers when companies need a license, but for a lay individual author, seeking legal advice and writing will not be an option.

Tools could (semi-)automatically do this task. To understand licenses, a tool needs to have the licenses in a machine-readable language. ccREL[6] is the language used by Creative Commons to represent their licenses. L4LOD[7] is a similar lightweight language. MPEG-21[8] is an ISO standard that permits the expression of licenses. ODRL[9] is a W3C standard made to represent complex types of policies. In this work we rely on the ODRL vocabulary for its capacity to express complex contextual constraints thanks to constraints and refinements. The ODRL vocabulary provides concepts that convey basic legal provision such as permission, duty and prohibition. The vocabulary targets the usage of content or services. A license can be considered as an ODRL Offer, an offer is composed of a list of rules and of an assigner, here it is the person licensing the resource. A rule is an abstract representation of a permission, a duty, or a prohibition, which we call status in this work. Each rule concerns one action in a status. As stated, ODRL allows expressing licenses with complex contexts thanks to *constraints* and *refinements*. An ODRL constraint allows to restrict a given rule to a specified context. While an ODRL refinement allows to further precise an action in a rule.

Figure 1 shows a graphical representation of three licenses (see in Appendix A their ODRL expression). L1 contains three rules (A, B, and C). Rule A permits the distribution of the target with a constraint (dateTime > 2030) and a refinement of the action Distribution (mediaType on TV and online streaming). Rule B obliges a compensation (payAmount of 1000$). And Rule C prohibits Commercial Use until 2050 in Europe. After 2050, this rule is not applicable and the user has to refer to standard copyright rules. L2 contains also three rules (D, E and F). Roughly speaking, with L2 the target can be distributed until 2050 on TV, online streaming, and movies (rule D). It can also be distributed without time constraint on printed medias as journals and books (rule E). But L2 has a prohibition of distribution on TV before 2040 (rule F).

The license that would protect a new resource combining resources protected by L1 and L2 should

[6]https://creativecommons.org/ns
[7]https://ns.inria.fr/l4lod/
[8]https://www.mpeg.org/standards/MPEG-21/
[9]https://www.w3.org/TR/odrl-model/

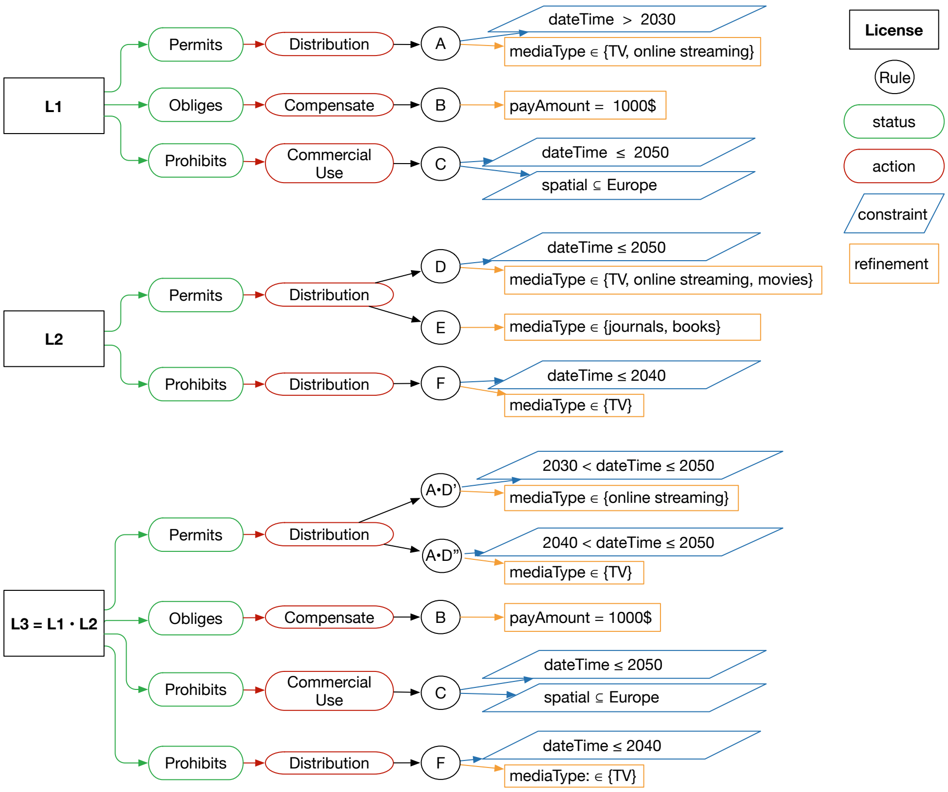

Figure 1: Graphical representation of three licenses with ODRL constraints in blue, and ODRL refinements in orange. License L3 is the composition of licenses L1 and L2, marked as L1•L2.

(1) contain every prohibition and duty of L1 and L2 (i.e., the prohibitions and duties of L1 and L2 are propagated), and (2) contain the intersection of the permissions of L1 and L2 that are not in contradiction with the propagated prohibitions and duties. If such a license exists, it must be equally or more restrictive than both L1 and L2. In our example, neither L1 nor L2 can be this license. We cannot determine which one is more restrictive because neither fully encompasses the other. Neither contains at least the same prohibitions and obligations as the other, nor does either include at most the same permissions as the other.

We argue that L3 in Figure 1 meets requirements (1) and (2). L3 is the least restrictive license that complies to L1 and L2 because, on the one hand, it includes all the prohibitions and duties propagated by L1 and L2 (and no more) with the minimal constraints and refinements necessary to ensure compliance with both licenses. On the other hand, it retains as many permissions as possible without contradicting the propagated prohibitions and obligations, while keeping constraints and refinements to a minimum. Concretely, L3 includes every duty and prohibition described by L1 and L2. It does not contain the permission E of L2, as there is no permission in L1 which intersects with the constraint and refinement of E. Finally, L3 contains the permissions A of L1 and D of L2 as the permissions A•D' and A•D". These two rules contain the intersection of the constraints and refinements of A and D, except what is stated as prohibited by F. Other licenses that would comply to L1 and L2 are licenses that are compliant with L3, but with additional constraints and refinements over the permissions than L3, and with more obligations and duties. This license would be more restrictive than L3. Indeed, following this idea, a lot of more restrictive licenses compliant to L1 and L2 may exist.

Obtaining L3 automatically is challenging and still an open problem [2], especially with complex licenses that contain constraints and refinements. If multiple works already tackle the challenge of license composition [1, 3, 4, 5, 6], not all can support contextual constraints [3, 4, 5, 6], and none of them fully supports them as described with ODRL constraints and refinements.

Thus, our problem statement is *how to automatically build a license that complies with a set of complex* *licenses containing constraints and refinements?* The challenge is to guarantee that such a license is the least restrictive license equally or more restrictive than the complex licenses it complies to. In this paper we propose CLiC (Complex License Composition), an approach to build a license by composing a set of complex licenses. The main idea of our approach is to order constraints and refinements based on their restrictiveness using a set-based comparison. This approach guarantees that the resulting license is the least restrictive license that can be used to cover a resource combining source materials protected by different complex licenses. A implementation of CLiC is available on Git[10].

The rest of the paper is structured as follows. We present existing works in license composition in Section 2. We formally define CLiC in Section 3. We present two use cases in Section 4, one with Creative Commons licenses, and one with the complex licenses of Figure 1. Then we discuss about the limitation and future works surrounding CLiC in Section 5. Finally, Section 6 concludes the paper.

2. Related works

CLiC is an approach to license composition that builds upon but also goes beyond existing work. For instance the study by Gangadharan et al. [3] is an approach specifically tailored for licenses in the context of services. It leverages a matchmaking algorithm to analyze the compatibility and enable the composition of service licenses. Gangadharan et al. utilize an extension of ODRL for services named ODRL/L(S) that they defined in [7]. This language contains elements of contextual constraints, such as the financial model of a service, or its warranties, indemnities, and limitation of liabilities. Despite that, ODRL/L(S) does not leverage the constraints and refinements present in the base language. Therefore, it cannot compose licenses with contextual constraints and refinements as described in ODRL.

The work by Mesiti et al. [5] is an approach made to help composing resources and to generate a license expressing the rights in the reuse for the resulting resource. It represents licenses using MPEG-21 as only a list of grants with conditions. This can be understood as licenses containing permissions with constraints. This approach does not consider prohibitions, duties, or refinements. The use of conditions allows Mesiti et al. to represent complex licenses by precising, for instance, time or spatial constraints. In addition, their approach can check whether a specific user's profile violates a license's conditions. But the representation of a license as a list of grants limits this work to licenses only composed of permissions, and conditions only allow the representation of constraints, not refinements.

The paper by Governatori et al. [4] uses deontic logic and defeasible logic to represent licenses and compose them into a new compliant license. The method does not allow the expression of contextual constraints and refinements as described with ODRL. Furthermore, when composing a new license from a set, this approach adds all prohibitions and duties from the set of licenses to the new license, potentially composing a license that contradicts itself and producing an invalid license.

DALICC [6] is a software framework that helps users in the reutilization of resources. DALICC allows for the user to search existing licenses, to create and check the coherency of new licenses, and to evaluate their equivalence or compatibility in regards to a library of well-known licenses. The framework uses answer set programming to detect potential conflicts between licenses. Status and actions are represented by DALICC as a knowledge graph that links together actions using their semantic relations. As shown in the current iteration of the tool[11], DALICC allows the creation of new licenses with the specification of temporal and spatial constraints. But, to the best of our knowledge, DALICC does not incorporate contextual constraints exhaustively, only singular examples. In addition, DALICC does not allow for the automated composition of multiple licenses in order to find or generate a new license that fits the users' situation. Therefore, DALICC cannot express or compose complex licenses such as the ones considered in this work.

The model CaLi proposed in [1] relies on the concept of restrictiveness. The model partially orders licenses on a lattice of restrictiveness, and using that lattice, define a compatibility relation between licenses. Despite the fact that the paper [1] does not explicitly specify it, the formalization of a lattice

[10] https://gitlab.univ-nantes.fr/clara/clic
[11] https://www.dalicc.net/license-composer/

of licenses allows for the composition of any pair of license thanks to the join operation[12]. But the model CaLi does not incorporate contextual constraints, in particular, it cannot trivially be extended to support ODRL refinements. This is due to the fact that CaLi restricts licenses to having only a single rule on a given action, while ODRL refinements foster the use of the same action multiple times with different refinements in the same license.

In conclusion, none of the existing works on license composition can compose the license L3 of our motivating use case. That is why we propose CLiC. As most of the approaches described in this section [1, 3, 4, 6], CLiC describes a license as a set of three types of rules, permission, duties, and prohibitions. As opposed to the studied methods, we leverage contextual constraints and refinements based on ODRL. Our approach proposes a composition of licenses based on the comparison of contextual constraints in terms of domains, i.e., we express constraints and refinements as domains of values. This representation as domain allows CLiC to incorporate any type of contextual constraint that could be represented as a domain. Our comparison is achieved inspired by the notion of restrictiveness as described in [1]. The idea is that, given two licenses, one is considered compliant with the other if it is equally or more restrictive than the latter.

3. CLiC: composing complex licenses with constraints and refinements

In this work we propose CLiC (Complex License Composition), a method to compose a set of complex licenses in order to build the least restrictive license compliant to that set. To achieve this, we base our approach on the notion of restrictiveness describe in [1]. As every rule in a complex license can contain multiple constraints and refinements, we compare constraints and refinements in terms of restrictiveness. For that, the main idea we propose is to consider constraints and refinements as sets of values, i.e., as domains. As a basic principle, we consider that a permission with a constraint is more restrictive than a permission without constraints, because a constraint restricts the permission to a specific context. On the contrary, a duty or an obligation with a constraint is less restrictive than a duty or obligation without constraints, because a constraint reduces the restrictiveness of the duty or the obligation to a specific context. This idea is analogous for the refinement of actions. Based on that principle and the hypothesis that constraints and refinements define domains, we can compare them based on the inclusion of those domains.

A complex license is a set of rules. Each rule concerns a status, a specific action, a collection of constraints, and a collection of refinements. Composing two complex licenses therefore devolves into comparing the rules of the original licenses to find the least restrictive set of rules that composes the new license, with the right constraints and refinements. If a new license can be composed from a set of source licenses, our set-based approach for comparing domains assures that the new license integrates every prohibition and duty contained in the source licenses, and that it contains as much permissions as possible, leading to the least restrictive license compliant to an original set of licenses.

In this section we describe CLiC, our approach to compose a set of licenses into a new license. Section 3.1 introduces the formal definitions that allow to compare constraints and refinements in the context of a status and an action. Section 3.2 introduces the algorithm that, based on these definitions, allows to obtain the least restrictive license that is compliant to the licenses it composes.

3.1. Formal comparison of constraints and refinements based on their restrictiveness

To compare the rules of different licenses, we need to be able to compare the constraints and refinements of different rules based on restrictiveness. This section introduces three definitions. Definition 1 formally defines the restrictiveness order over collections of constraints while Definition 2 formally defines the restrictiveness order over collections of refinements. With these definitions we can finally define complex licenses in Definition 3.

[12] https://en.wikipedia.org/wiki/Join_and_meet

In the following, we formalize a constraint and a collection of constraints as a set of constraints of distinct types, e.g., a temporal constraint and a spatial constraint. Limiting collections to have constraints of distinct types allows for an easier comparison of two collections. In addition, it does not reduce the variety of licenses that can be expressed, because two constraints of a same type can be expressed as a single constraint by utilizing the intersection of their domains, e.g., the two constraints (dateTime < 2050) and (dateTime $\geq$ 2030) can be joined as (2030 $\leq$ dateTime < 2050).

After defining collections of constraints, we also formalize the restrictiveness relation between collections in the context of a status.

Definition 1 (Restrictiveness order over collections of constraints).

Constraint. *We define a constraint c in the context of a given status as a pair $\left(c^{type}, c^{domain}\right)$ where the first element is the type of the constraint and the second element is the domain of the constraint. We name $\mathcal{C}_s$ the set of all possible constraints in the status s. In ODRL, c^{type} corresponds to* leftOperand *and c^{domain} corresponds to a combination of the* Operator *and* rigthOperand.

Collection of constraints. *We define a collection of constraints O as a set of constraints, each with a different type. We name O^{types} the list of types present in a collection of constraints. We formally define all the possible collections of constraints $\mathcal{O}_s$ as a subset of the powerset of $\mathcal{C}_s$, $\wp\left(\mathcal{C}_s\right)$, given a status s:*

$$\mathcal{O}_s = \left\{O \in \wp\left(\mathcal{C}_s\right) | \forall c_i, c_j \in O, c_i^{type} = c_j^{type} \implies c_i = c_j\right\} \quad (1)$$

Restrictiveness relation over the set of constraints $\mathcal{C}_s$. *We consider that, (i) for the status Prohibition and Duty, given two constraints $c_i, c_j \in \mathcal{C}_s$, the former constraint is less restrictive than the latter in the context of a same status, $c_i \leq_s c_j$, iff $c_i^{type} = c_j^{type} \wedge c_i^{domain} \subseteq c_j^{domain}$. And (ii) for the status Permission, $c_i \leq_s c_j$ iff $c_i^{type} = c_j^{type} \wedge c_i^{domain} \supseteq c_j^{domain}$. For instance, this allows to establish a restrictiveness relation for the constraints of C and F in Figure 1 as (dateTime $\leq$ 2040) $\leq_{Prohibition}$ (dateTime $\leq$ 2050).*

Restrictiveness relation over the set of collections $\mathcal{O}_s$. *From that relation over constraints, we define a restrictiveness relation $\leq_s$ over $\mathcal{O}_s$ by comparing pairwise the domains of the constraints that have the same type in the context of the same status s. The intuition is that for every status, the collection whose constraints are always equally or less restrictive than the constraints of another collection is equally or less restrictive than that other collection.*

For $s \in \{Duty, Prohibition\}$:

$$\forall O_i, O_j \in \mathcal{O}_s, O_i \leq_s O_j \iff O_i^{types} \subseteq O_j^{types} \wedge \left(\forall c_i \in O_i, \exists c_j \in O_j, \text{ such that } c_i \leq_s c_j\right) \quad (2)$$

For $s \in \{Permission\}$:

$$\forall O_i, O_j \in \mathcal{O}_s, O_i \leq_s O_j \iff O_i^{types} \supseteq O_j^{types} \wedge \left(\forall c_j \in O_j, \exists c_i \in O_i, \text{ such that } c_i \leq_s c_j\right) \quad (3)$$

For practicality, we also redefine two set operators over collections of constraints: the intersection, and the subtraction. The operators are defined using the equality of the type of the constraints from two collections, and the intersection or subtraction of the constraints' domains. The intersection, noted as $\cap$, is the intersections of all the constraints in two collections, defined as follows: for $O_i, O_j \in \mathcal{O}_s$, $O_i \cap O_j = \{c \in \mathcal{C}_s | \exists c_i \in O_i, \exists c_j \in O_j, \left(c^{type} = c_i^{type} = c_j^{type}\right) \wedge \left(c^{domain} = c_i^{domain} \cap c_j^{domain}\right)\}$. And the subtraction, noted as $-$, is a collection where the domains of its constraints have been subtracted by the domains of the constraints of another collection. It is obtained by performing the following operation: for two collections of constraints $O_i, O_j \in \mathcal{O}_s$, for every constraint $c_i \in O_i$, if there exists a constraint $c_j \in O_j$ with the same type as c_i, then $\left(c_i^{type}, c_i^{domain} \setminus c_j^{domain}\right)$ is one of the constraints of $O_i - O_j$, otherwise, c_i is one of the constraints of $O_i - O_j$.

An example of a constraint is $(\text{dateTime}, \{x | x > 2050\})$, it limits the dates on which a rule is applied to every dates that comes after the year 2025. And an example of a collection of two constraints is

$\{(\text{dateTime}, \{x|x > 2050\}), (\text{spatial}, \{x|x \subseteq \text{Europe}\})\}$, in addition of the previous constraint, it limits the place targeted by the rule to be within Europe.

We define a refinement, a collection of refinements, and the restrictiveness relation between collections of refinements in the same way. The difference between constraints and refinements is that a constraint is defined in the context of a status, while a refinement is defined in the context of a status and of an action. That is because a refinement redefines an action to a more specific application.

Definition 2 (Restrictiveness order over collections of refinements).

Refinement. *We define a refinement* r *in the context of a given status and of a specific action as a pair* $\left(r^{type}, r^{domain}\right)$ *where the first element is the type of the refinement and the second element denotes the domain of the refinement. We name* $\mathcal{R}_{s,a}$ *the set of all possible refinements for the status* s *and action* a*.*

Collection of refinements. *We define a collection of refinements* E *as a set of refinements, each with a different type. We name* E^{types} *the list of types present in a collection of refinements* E*. We formally define all the possible collection of refinements for a given status* s *and a given action* a *as* $\mathcal{E}_{s,a}$ *:*

$$\mathcal{E}_{s,a} = \{E \in \wp(\mathcal{R}_{s,a}) \,|\forall r_i, r_j \in E, r_i^{type} = r_j^{type} \implies r_i = r_j\} \tag{4}$$

Restrictiveness relation over the set of refinements $\mathcal{R}_{s,a}$**.** *As with constraints we define the restrictiveness relation between two refinements. On a given action, for the status Prohibition and Duty, for two refinements* $r_i, r_j \in \mathcal{R}_{s,a}$*, then* $r_i \leq^r_{s,a} r_j$ *if* $r_i^{type} = r_j^{type} \wedge r_i^{domain} \subseteq r_j^{domain}$*. And for the status Permission,* $r_i \leq^r_{s,a} r_j$ *if* $r_i^{type} = r_j^{type} \wedge r_i^{domain} \supseteq r_j^{domain}$*. For instance, this allows us to establish a restrictiveness order for the refinements of D and A in Figure 1 by saying that* $(mediaType \in \{TV, onlinestreaming, movie\}) \leq^r_{Permission,Distribution} (mediaType \in \{TV, onlinestreaming\})$*.*

Restrictiveness relation over the set of collections $\mathcal{E}_{s,a}$**.** *From that relation over refinements, we define a restrictiveness relation order* $\leq_{s,a}$ *over* $\mathcal{E}_{s,a}$ *by comparing pairwise the domains of the refinements that have the same type in the context of the same status* s *and action* a*. The intuition is that for every status paired with an action, a first collection where the refinements are always equally or less restrictive than the refinements of a second collection is then equally or less restrictive than that second collection.*

For $s \in \{Duty, Prohibition\}$*:*

$$\forall E_i, E_j \in \mathcal{E}_{s,a}, E_i \leq_{s,a} E_j \iff E_i^{types} \subseteq E_j^{types} \wedge \left(\forall r_i \in E_i, \exists r_j \in r_j, r_i \leq^r_{s,a} r_j\right) \tag{5}$$

For $s \in \{Permission\}$*:*

$$\forall E_i, E_j \in \mathcal{E}_{s,a}, E_i \leq_{s,a} E_j \iff E_i^{types} \supseteq E_j^{types} \wedge \left(\forall r_j \in E_j, \exists r_i \in E_i, r_i \leq^r_{s,a} r_j\right) \tag{6}$$

Additionally, in the same way as for two collections of constraints, we define the intersection of two collection of refinements. The intersection is defined as follows: for $E_i, E_j \in \mathcal{E}_{s,a}$*,* $E_i \cap E_j = \{r \in \mathcal{R}_{s,a} | \exists r_i \in E_i, \exists r_j \in E_j, \left(r^{type} = r_i^{type} = r_j^{type}\right) \wedge \left(r^{domain} = r_i^{domain} \cap r_j^{domain}\right)\}$*. The subtraction, noted* $-$*, is obtained by performing the following operation: for two collections of refinements* $E_i, E_j \in \mathcal{E}_{s,a}$*, for every refinement* $r_i \in E_i$*, if there exists a refinement* $r_j \in E_j$ *with the same type as* r_i*, then* $\left(r_i^{type}, r_i^{domain} \setminus r_j^{domain}\right)$ *is a refinement of* $E_i - E_j$*, otherwise,* r_i *is a refinement of* $E_i - E_j$*.*

An example of a refinement on the action distribute is $(\text{mediaType}, \{\text{TV}\})$, which redefines the action to only concern distribution on television. And an example of a collection of refinements on the same action would be $\{(\text{mediaType}, \{\text{TV}\}), (\text{language}, \{\text{en-US}\})\}$, which redefines the action to only concern distribution on TV and in American English.

With the definition of a collection of constraints and a collection of refinements, it is possible to define a complex licenses as a set of rules. Each rule contains a status and an action, and that rule can be limited by a collection of constraints and a collection of refinements.

Definition 3 (Complex license).

Rule. *Let $\mathcal{S}$ be a set of status, $\mathcal{A}$ be a set of actions, $\mathcal{O}_s$ be a set of collections of constraints for a status s, and $\mathcal{E}_{s,a}$ be a set of collections of refinements for a status s and an action a. A rule is a tuple (s, a, O, E) defined on $\mathcal{S} \times \mathcal{A} \times \mathcal{O}_s \times \mathcal{E}_{s,a}$. The collections of constraints and refinements should be understood as a conjunction. That means that all the constraints and refinements present in a rule need to be satisfied, otherwise the rule cannot be exercised. For a rule r (s, a, O, E), we name respectively $r.s$, $r.a$, $r.O$, and $r.E$ each of its element.*

Complex license. *A license l is a set of rules, each defined on $\mathcal{S} \times \mathcal{A} \times \mathcal{O}_s \times \mathcal{E}_{s,a}$. Note that $\emptyset \in \mathcal{O}_s$ and $\emptyset \in \mathcal{E}_{s,a}$, meaning that it is possible for a rule in a complex license to have no refinements and no constraints. We denote by l_a^s the set of rules assigned to the action a with the status s in the license l, e.g., $l_{Derive}^{Permission}$ is the list of rules in the license l relating to the status $Permission$ and to the action $Derive$.*

For example, the license L1 given in Figure 1 can be expressed as follows:

$$
\begin{aligned}
L1\colon \{ & (\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|x > 2030\})\}, \{(\text{mediaType}, \{\text{TV}, \text{online streaming}\})\}), \\
& (\text{Duty}, \text{Compensate}, \emptyset, \{(\text{payAmount}, \{1000\$\})\}), \\
& (\text{Prohibition}, \text{Commercial Use}, \{(\text{dateTime}, \{x|x \leq 2050\}), (\text{spatial}, \{x|x \subseteq \text{Europe}\})\}, \emptyset) \ \}
\end{aligned}
$$

3.2. An algorithm to compose complex licenses

With the ability to compare constraints and refinements in the context of a status and an action, it is possible to compare the rules of a set of complex licenses. By comparing and composing the rules of two complex licenses, we can obtain the least restrictive license that is compliant to that pair. For that, we propose the Algorithm 1.

Algorithm 1 describes the function compose which, from two complex licenses l_i and l_j, returns the least restrictive license compliant to l_i and l_j or an empty set if it is not possible. To achieve that we perform four steps for every action present in l_i or l_j. We have as hypothesis that the licenses l_i and l_j are correct and do not contain any rule preventing re-licensing (e.g., the obligation Share Alike, or the prohibition of Derivative Works).

Step 1. Propagation of Duties and Prohibitions. Line 5 adds every prohibition and duty defined for a specific action in l_i or l_j to the resulting license. This guarantees that the resulting license will be compliant to l_i and l_j in regards to the prohibitions and duties. Additionally, as we keep the constraints and refinements within the prohibitions and duties, we ensure that the resulting license is as not restrictive as possible. Lines 6 to 10 remove from the resulting license every unnecessary prohibition or duty. That is, every prohibition or duty whose constraints and refinements are equally or less restrictive than another prohibition or duty (cf. Definition 1 and 2). This effectively simplifies the resulting license by removing rules that are contained in another rule.

Step 2. Detection of incompatible Duty and Prohibition. Lines 11 to 13 detect if a couple of propagated duty and prohibition makes the composed license incoherent. If that is the case, compose returns an empty set because the licenses are incompatible and a compliant license cannot exist.

Step 3. Addition of the intersection of Permissions. For each pair of permissions, one coming from l_i and one from l_j, the lines 14 to 16 compose a new permission and add it to the resulting license. The created permission concerns the same action as the original pair of permissions, and it contains the intersections of the collections of constraints and of refinements from that original pair. By using the intersection of the collections, we ensure that the created permission will be as not restrictive as possible while still being equally or more restrictive than the original pair.

Step 4. Ensuring coherency of Permissions vis-à-vis the Duties and Prohibitions. Lines 17 to 24 ensure the coherency of the domains of Permissions vis-à-vis the domains of Duties and Prohibitions by removing the values of the domain of each permission that contradicts a prohibition or duty. For that, our approach consists of only keeping subsets of each permission. Lines 18 to 24 keep the subsets of each permission which do not contradict a prohibition or duty, those correspond to the parts of which the collections of refinements or constraints do not intersect with the collections of refinement or constraints of the prohibition or duty. We only add back the parts with non-empty collections. Lines

Algorithm 1: Compose a pair of complex licenses.

1 **Function** *compose*(l_i, l_j)**:**
  Data:
  l_i: A complex license defined on the set of actions $\mathcal{A}_i$,
  l_j: A complex license defined on the set of actions $\mathcal{A}_j$.
  Precondition : l_i and l_j do not contain any rule preventing re-licensing, e.g., ShareAlike.
  Result: The least restrictive complex license compliant to both l_i and l_j.
2   $new\_license \leftarrow \emptyset$
3   **forall** $action \in \mathcal{A}_i \cup \mathcal{A}_j$ **do**
    // Step 1. Add every rule defined with the status Prohibition and Duty.
4   **forall** $status \in \{Prohibition, Duty\}$ **do**
5       $new\_license \leftarrow new\_license \cup l_{i\,action}^{status} \cup l_{j\,action}^{status}$
      // Remove unnecessary rules defined with the status Prohibition and Duty.
6       $unnecessary\_rules \leftarrow \emptyset$
7   **forall** $(new\_rule, new\_rule') \in new\_license_{action}^{status} \times new\_license_{action}^{status}$ **do**
8   **if** $new\_rule.O \leq_s new\_rule'.O$ **and** $new\_rule.E \leq_{s,a} new\_rule'.E$ **and** $new\_rule \neq new\_rule'$ **then**
9           $unnecessary\_rules \leftarrow unnecessary\_rules \cup \{new\_rule\}$
10       $new\_license_{action}^{status} \leftarrow new\_license_{action}^{status} \setminus unnecessary\_rules$
    // Step 2. The licenses are incompatible if a rule with the status Prohibition overlaps with a rule with the status Duty on a same action
11   **forall** $(duty, prohib) \in new\_license_{action}^{Duty} \times new\_license_{action}^{Prohibition}$ **do**
12   **if** $duty.O \cap prohib.O \neq \emptyset$ **and** $duty.E \cap prohib.E \neq \emptyset$ **then**
13   **return** $\emptyset$
    // Step 3. Add the rules of l_i and l_j which are defined with the status Permission, with the intersection of their collections.
14   **forall** $(rule_i, rule_j) \in l_{i\,action}^{Permission} \times l_{j\,action}^{Permission}$ **do**
15   **if** $rule_i.O \cap rule_j.O \neq \emptyset$ **and** $rule_i.E \cap rule_j.E \neq \emptyset$ **then**
16         $new\_license \leftarrow new\_license \cup \{(Permission,\ action,\ rule_i.O \cap rule_j.O,\ rule_i.E \cap rule_j.E)\}$
    // Step 4. Remove every part of a rule defined with the status Permission that overlaps with a rule defined with the status Prohibition or Duty.
17   **forall** $rule \in new\_license_{action}^{Duty} \cup new\_license_{action}^{Prohibition}$ **do**
      // Keep only the subsets of each *perm* that do not overlap with *rule*.
18       $perms\_subsets \leftarrow \emptyset$
19   **forall** $perm \in new\_license_{action}^{Permission}$ **do**
20   **if** $perm.E - rule.E \neq \emptyset$ **then**
21           $perms\_subsets \leftarrow perms\_subsets \cup \{(Permission,\ action,\ perm.O,\ perm.E - rule.E)\}$
22   **if** $perm.O - rule.O \neq \emptyset$ **and** $perm.E \cap rule.E \neq \emptyset$ **then**
23           $perms\_subsets \leftarrow perms\_subsets \cup \{(Permission,\ action,\ perm.O - rule.O,\ perm.E \cap rule.E)\}$
24       $new\_license_{action}^{Permission} \leftarrow perms\_subsets$
25   **return** $new\_license$

20 and 21 keeps the subset of the permission that has different refinements from the prohibition or duty, without changing its constraint. This corresponds to the collections of refinements of the prohibition or duty subtracted to the collection of refinements of the permission (cf. Definition 2). Then, Lines 22 and 23 keeps the subset of the permission that has the same refinements as the prohibition or duty, but different constraints. This corresponds to the intersection of the collections of refinements (cf. Definition 2) and to the collection of constraints of the permission minus the collection of constraints

$$\text{CC BY}: \{ (\text{Permission}, \text{Distribution}, \emptyset, \emptyset), \\ (\text{Permission}, \text{Reproduction}, \emptyset, \emptyset), \\ (\text{Permission}, \text{Sharing}, \emptyset, \emptyset), \\ (\text{Permission}, \text{Derivative Works}, \emptyset, \emptyset), \\ (\text{Permission}, \text{Commercial Use}, \emptyset, \emptyset), \\ (\text{Duty}, \text{Notice}, \emptyset, \emptyset), \\ (\text{Duty}, \text{Attribution}, \emptyset, \emptyset) \}$$

$$\text{CC BY-NC}: \{ (\text{Permission}, \text{Distribution}, \emptyset, \emptyset), \\ (\text{Permission}, \text{Reproduction}, \emptyset, \emptyset), \\ (\text{Permission}, \text{Sharing}, \emptyset, \emptyset), \\ (\text{Permission}, \text{Derivative Works}, \emptyset, \emptyset), \\ (\text{Duty}, \text{Notice}, \emptyset, \emptyset), \\ (\text{Duty}, \text{Attribution}, \emptyset, \emptyset), \\ (\text{Prohibition}, \text{Commercial Use}, \emptyset, \emptyset) \}$$

Figure 2: Complex licenses representation of CC BY and CC BY-NC.

of the prohibition or duty (cf Definition 1). Finally, on Line 24, the subsets of each Permission replace the Permissions of the new license.

4. Use cases of license composition

This section illustrates our approach with the composition of two pairs of licenses, one simple case and one complex case. The simple use case concerns Creative Commons licenses, which do not contain constraints or refinements. It is developed in Section 4.1. The complex use case is the one shown in Figure 1, it explores most steps of Algorithm 1. It is developed in Section 4.2.

4.1. Use case with Creative Commons licenses

A simple example, such as the composition of Creative Commons licenses, helps showcasing the soundness of Algorithm 1. We describe the composition of CC BY[13] and CC BY-NC[14] whose composition produces CC BY-NC. That is because CC BY-NC is compliant to CC BY.

The expressions of CC BY and CC BY-NC following Definition 3 are shown in Figure 2. The license CC BY permits the five actions Distribution, Reproduction, Sharing, Derivative Works, and Commercial Use. It obliges the actions Notice and Attribution, and it prohibits none. The license CC BY-NC permits the four actions Distribution, Reproduction, Sharing, and Derivative Works. It obliges the actions Notice and Attribution, and it prohibits the action Commercial Use.

When composing licenses with no contextual constraints or refinements, Algorithm 1 keeps every duty and prohibition present in at least one license, and every permission present in both licenses. Step 1 of Algorithm 1 will add to the new license the obligations of Notice and Attribution, originating from the two licenses as well as the prohibition of Commercial Use. Step 3 adds permissions contained in both CC BY and CC BY-NC. Steps 2 and 4 of the algorithm have no impact in this example. The resulting composed license is perfectly equivalent to CC BY-NC.

4.2. Complex use case with constraints and refinements

This section explains how CLiC obtains the license L3 of Figure 1 using Algorithm 1. We first describe the licenses L1 and L2 from Figure 1 as complex licenses (cf. Definition 3). The license L1 corresponds to the following complex license. We name the rules of L1 respectively A, B, and C according to Figure 1.

$$L1: \{ (\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|x > 2030\})\}, \{(\text{mediaType}, \{\text{TV}, \text{online streaming}\})\}), \#\mathbf{A} \\ (\text{Duty}, \text{Compensate}, \emptyset, \{(\text{payAmount}, \{1000\$\})\}), \#\mathbf{B} \\ (\text{Prohibition}, \text{Commercial Use}, \{(\text{dateTime}, \{x|x \leq 2050\}), (\text{spatial}, \{x|x \subseteq \text{Europe}\})\}, \emptyset) \ \#\mathbf{C} \}$$

The license L2 corresponds to the following complex license. We name the rules of L2 D, E, and F.

$$L2: \{ (\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|x \leq 2050\})\}, \{(\text{mediaType}, \{\text{TV}, \text{online streaming}, \text{movies}\})\}), \#\mathbf{D} \\ (\text{Permission}, \text{Distribution}, \emptyset, \{(\text{mediaType}, \{\text{journal}, \text{books}\})\}), \#\mathbf{E} \\ (\text{Prohibition}, \text{Distribution}, \{(\text{dateTime}, \{x|x \leq 2040\})\}, \{(\text{mediaType}, \{\text{TV}\})\}) \ \#\mathbf{F}\}$$

[13] https://creativecommons.org/licenses/by/4.0/deed.en
[14] https://creativecommons.org/licenses/by-nc/4.0/deed.en

In the following paragraphs, we compose the complex licenses L1 and L2 following Algorithm 1.

Step 1 adds every duty and prohibition from L1 and L2 into the new license, this leads to the addition of three rules, the duty B and the prohibition C from L1 as well as the prohibition F from L2. As there is no duty or prohibition contained in another prohibition or duty, we keep every rule in the new license.

Step 2 verifies if the composition is possible. It verifies that no pair of rules defined on a same action, one with the status Duty and one with the status Prohibition, have intersecting collections of constraints and refinements. This example has no issue because the rule B is defined on the action Compensate, while the rules C and F are defined on the actions Commercial Use and Distribution.

Step 3 composes rules A and D, and rules A and E, but the latter has no result. On the one hand, the combination of rules A and E does not lead to a new permission as their collections of refinements have no intersection, i.e., $(\text{mediaType}, \{\text{TV}, \text{online streaming}\}) \cap (\text{mediaType}, \{\text{journals}, \text{books}\}) = \emptyset$. On the other hand, the combination of rule A in L1 and rule D in L2 leads to the new rule A•D. This new rule is defined on the status Permission, the action Distribution, and contains the intersection of the collections of both original permissions as its collections of constraints and refinements. Here, the intersection of the collections of constraints is $\{(\text{dateTime}, \{x|x > 2030\})\} \cap \{(\text{dateTime}, \{x|x \leq 2050\})\}$, which is equal to the collection: $\{(\text{dateTime}, \{x|2030 < x \leq 2050\})\}$. And the intersection of the collections of refinements is $\{(\text{mediaType}, \{\text{TV}, \text{online streaming}\})\} \cap \{(\text{mediaType}, \{\text{TV}, \text{online streaming}, \text{movies}\})\}$, which is equal to the collection $\{(\text{mediaType}, \{\text{TV}, \text{online streaming}\})\}$. Therefore, this step adds (temporally) the rule A•D = $(\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|2030 < x \leq 2050\})\}, \{(\text{mediaType}, \{\text{TV}, \text{online streaming}\})\})$.

Step 4 ensures that the permissions of the new license do not contradict its prohibitions or duties. Therefore, rule A•D is compared with duty B and prohibitions C and F. The comparison of rule A•D with B and C does not raise any issue as A•D is defined on the action Distribution, B on the action Compensate, and C on the action Commercial Use. The comparison of rules A•D and F is more complex as both are defined on the action Distribution. Therefore, A•D is removed from the new license and two new rules are created, named A•D' and A•D". A•D' and A•D" are versions of A•D with truncated collections of constraints and refinements that prevent the potential contradiction with the prohibition F. Following the formulas of Algorithm 1, the two rules are the following: A•D' = $(\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|2030 < x \leq 2050\})\}, \{(\text{mediaType}, \{\text{online streaming}\})\})$, and A•D" = $(\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|2040 < x \leq 2050\})\}, \{(\text{mediaType}, \{\text{TV}\})\})$.

After the four steps, the new license created corresponds to the following complex license, which equals the license L3 in Figure 1.

$$
\begin{aligned}
L3\colon \{ & (\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|2030 < x \leq 2050\})\}, \{(\text{mediaType}, \{\text{online streaming}\})\}), \#\mathbf{A}\cdot\mathbf{D}' \\
& (\text{Permission}, \text{Distribution}, \{(\text{dateTime}, \{x|2040 < x \leq 2050\})\}, \{(\text{mediaType}, \{\text{TV}\})\}), \#\mathbf{A}\cdot\mathbf{D}'' \\
& (\text{Duty}, \text{Compensate}, \emptyset, \{(\text{payAmount}, \{1000\$\})\}), \#\mathbf{B} \\
& (\text{Prohibition}, \text{Commercial Use}, \{(\text{dateTime}, \{x|x \leq 2050\}), (\text{spatial}, \{x|x \subseteq \text{Europe}\})\}, \emptyset), \#\mathbf{C} \\
& (\text{Prohibition}, \text{Distribution}, \{(\text{dateTime}, \{x|x \leq 2040\})\}, \{(\text{mediaType}, \{\text{TV}\})\})\ \#\mathbf{F}\ \}
\end{aligned}
$$

5. Discussion

As described, CLiC will allow for data re-use by providing a way to automatically compose new licenses in order to cover a material derived from different sources (see Figure 3, left part). From a lay person's point of view, such an automatisation of the license's composition based upon various licenses with complexes constraints that defer from one another will offer a relief. As the analysis and composition effort is handled by the algorithm, CLiC is an incentive for data-sharing. In this regard, our research offers promising perspectives for digital rights management in complex data-sharing scenarios.

The next step of such a work will be to translate the machine-readable composed license into natural language (see Figure 3, right part). Indeed, so far the composed license is offered in a machine-readable format but, in order to reach maximum utility, the license will need to be translated in a natural language in order not to condition its reading and application to the use of complex computer means. In this regards, it seems only logical to convey a contractual document, such as a license destined to human

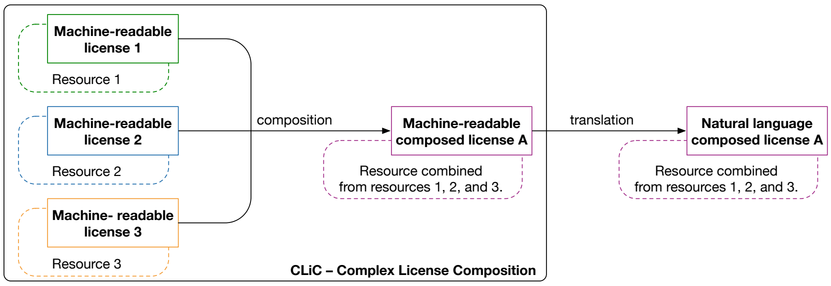

Figure 3: From a machine-readable license to a human-readable one

users, into a language that humans can make sens of. From a legal point of view, this human-readable license will be a prerequisite to demonstrate the license's existence in the case of a legal action as trying to enforce in court a machine-readable version of the license will prove to be complex and will question its validity. However, this essential task of translation, from machine-readable to natural language can also be computerised as well, using the ODRL vocabulary[15]. Indeed, the definitions attached to each ODRL concept can be retrieved in order to draft the natural language's license with the selected constraints. As RDF is structured in triples (subjects predicate object), defining simple phrases from a machine-readable license in RDF is possible. Indeed, it would be possible to define a template in English using the traditional structure of a license, beginning with a preamble, followed by definitions that clarify the terms and actions used. Next, the permissions, prohibitions, and obligations could be outlined, and the template may conclude with the conditions under which the license may be revoked and the legal consequences of non-compliance.

Some limits of such an enterprise have to be acknowledged. Indeed this type of automatic natural language drafting of a license will create a very simple and direct legal document which will be very different from what one would have expected from a license. In other words, the "drafting style" will constitute a limit. For instance, if the new composed license prohibits the commercial use of the resource, the drafting might read: "Prohibition: the *licensee* has the inability to exercise *commercial use* over the *resource*" as the definition of a prohibition, under ODRL, is "The inability to exercise an Action over an Asset". Such a drafting is understandable and will be enforced in a court of law. However, some users might want to correct the style and grammar in order to obtain a more human-readable friendly version. Moreover, one might want to adapt the text by translating it into another language such as French or German in order to adapt the license to its domestic context. This translation of the translation is not a legal prerequisite in order to enforce a license but might appear useful in order to disseminate the covered resource as the targeted re-users might better understand the text in her/his native language. Such translation from one natural language to another will not be provided by ODRL as the vocabulary is only published in English. With regards to those various limits, the user will be invited to pay attention to the human-readable document provided for, warning that the license need some improvement and proof-reading.

To facilitate the composed license's proof-reading, the next step, for the user, could be to use a Large Language Model (LLM). Such tool can easily re-word the composed license to make-it more human-friendly by generating the necessary and appropriate words. Here, again, the limits of this process should be acknowledged. The first will be that depending on the LLM used and its training data, the generated text's quality will vary a lot. Indeed, if the LLM has mostly been trained on English data, then it will perform better on a license drafted in English than on its translated versions in other languages. Additionally, cautions should be taken when using LLMs to take into account potential risk of harm. For instance, the composed license provided to the LLM should not include any personal data, as those generative AI hardly comply with data protection regulations due to a series of technical limitations that include the memorisation of personal data and their possible unwanted regurgitation.

[15]https://www.w3.org/TR/odrl-vocab/

Moreover, the user should monitor the generated content to ensure that no license's term has been modified and that the added text does not modify the intended meaning of the license. Therefore, human proof-reading and verification remain the essential last step of the process in order to ensure the license's correctness and, where appropriate and needed, a last review by a legal counsel. Such next steps will raise additional questions outside of the real of license composition such as in terms of the usability and the explainability of a license.

6. Conclusion

In this work, we propose CLiC, an approach to compose complex licenses that include constraints and refinements, as defined by ODRL. The core idea of our approach is to order constraints and refinements based on their restrictiveness using a set-based comparison. This method ensures that the resulting license is the least restrictive one capable of protecting a resource that combines resources protected by different complex licenses. Our contribution includes a set of definitions that formalize our approach, as well as an algorithm to facilitate its implementation.

CLiC aims to simplify and encourage the reuse and publication of licensed resources. It does not intend to provide legal advice but rather to offer a practical solution for users who may not seek such advice yet wish to combine various resources and generate an appropriate license. CLiC provides a tool that ensures the composed complex license does not infringe upon the licenses of the reused resources. By alleviating the burden of extensive legal analysis and the comparison of multiple licenses, CLiC introduces an innovative perspective as a data-sharing management tool. A future direction of this work is to define a methodology for translating machine-readable licenses into human-readable ones. Such a process could leverage the ODRL vocabulary, LLMs to enhance readability, and human verifications.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT to improve grammar and for spelling.

References

[1] B. Moreau, P. Serrano-Alvarado, M. Perrin, E. Desmontils, Modelling the Compatibility of Licenses, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. Gray, V. Lopez, A. Haller, K. Hammar (Eds.), The Semantic Web, Springer International Publishing, Cham, 2019, pp. 255–269. doi:10.1007/978-3-030-21348-0_17.

[2] S. Kirrane, S. Villata, M. d'Aquin, Privacy, security and policies: A review of problems and solutions with semantic web technologies, Semantic Web 9 (2018) 153–161. URL: https://journals.sagepub.com/doi/full/10.3233/SW-180289. doi:10.3233/SW-180289.

[3] G. R. Gangadharan, M. Weiss, V. D'Andrea, R. Iannella, Service License Composition and Compatibility Analysis, in: D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, B. J. Krämer, K.-J. Lin, P. Narasimhan (Eds.), Service-Oriented Computing – ICSOC 2007, volume 4749, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 257–269. URL: http://link.springer.com/10.1007/978-3-540-74974-5_21. doi:10.1007/978-3-540-74974-5_21, series Title: Lecture Notes in Computer Science.

[4] G. Governatori, A. Rotolo, S. Villata, F. Gandon, One License to Compose Them All, in: H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, K. Janowicz (Eds.), The Semantic Web – ISWC 2013, Springer, Berlin, Heidelberg, 2013, pp. 151–166. doi:10.1007/978-3-642-41335-3_10.

[5] M. Mesiti, P. Perlasca, S. Valtolina, On the Composition of Digital Licenses in Collaborative Environments, in: H. Decker, L. Lhotská, S. Link, J. Basl, A. M. Tjoa (Eds.), Database and Expert Systems Applications, Springer, Berlin, Heidelberg, 2013, pp. 428–442. doi:10.1007/978-3-642-40285-2_37.

[6] T. Pellegrini, V. Mireles, S. Steyskal, O. Panasiuk, A. Fensel, S. Kirrane, Automated Rights Clearance Using Semantic Web Technologies: The DALICC Framework, in: T. Hoppe, B. Humm, A. Reibold (Eds.), Semantic Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018, pp. 203–218. URL: http://link.springer.com/10.1007/978-3-662-55433-3_14. doi:10.1007/978-3-662-55433-3_14.

[7] G. Gangadharan Ramakrishnan, V. D'Andrea, R. Iannella, M. Weiss, ODRL/L(S): A Language for Service Licensing (2007). URL: https://iris.unitn.it/handle/11572/357868, accepted: 2023-12-01T11:51:08Z Publisher: Trento.

A. Appendix

The three licenses of Figure 1 in ODRL using the Turtle syntax.

Listing 1: License L1 expressed with ODRL.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix cc: <http://creativecommons.org/ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix : <http://example.com/> .

:L1 a odrl:Offer ;
    odrl:permission [
        rdfs:label "A" ;
        a odrl:Permission;
        odrl:action [
            rdf:value cc:Distribution ;
            odrl:refinement [
                odrl:leftOperand odrl:mediaType ;
                odrl:operator odrl:isAnyOf ;
                odrl:rightOperand ("TV"@en "online streaming"@en) ] ];
        odrl:constraint [
            odrl:leftOperand odrl:dateTime ;
            odrl:operator odrl:gt ;
            odrl:rightOperand "2030-12-31"^^xsd:date ]
    ] ;
    odrl:obligation [
        rdfs:label "B" ;
        a odrl:Duty;
        odrl:action [
            rdf:value odrl:compensate ;
            odrl:refinement [
                odrl:leftOperand odrl:payAmount ;
                odrl:operator odrl:eq ;
                odrl:rightOperand "1000"^^xsd:decimal ;
                odrl:unit <https://dbpedia.org/page/United_States_dollar> ] ]
    ];
    odrl:prohibition [
        rdfs:label "C" ;
        a odrl:Prohibition;
        odrl:action cc:CommercialUse ;
        odrl:constraint [
            a odrl:LogicalConstraint ;
            odrl:and ([
                    odrl:leftOperand odrl:dateTime ;
                    odrl:operator odrl:lteq ;
                    odrl:rightOperand "2050-12-31"^^xsd:date
                ] [
                    odrl:leftOperand odrl:spatial ;
                    odrl:operator odrl:isPartOf ;
                    odrl:rightOperand <https://dbpedia.org/page/Europe> ] ) ]
    ] .
```

Listing 2: License L2 expressed with ODRL.

```
:L2 a odrl:Offer ;
    odrl:permission [
        rdfs:label "D" ;
        a odrl:Permission;
        odrl:action [
            rdf:value cc:Distribution ;
            odrl:refinement [
                odrl:leftOperand odrl:mediaType ;
                odrl:operator odrl:isAnyOf ;
                odrl:rightOperand ("TV"@en "online streaming"@en "movies"@en) ] ] ;
        odrl:constraint [
            odrl:leftOperand odrl:dateTime ;
            odrl:operator odrl:lteq ;
            odrl:rightOperand "2050-12-31"^^xsd:date ]
    ] , [
        rdfs:label "E" ;
        a odrl:Permission;
        odrl:action [
            rdf:value odrl:compensate ;
            odrl:refinement [
                odrl:leftOperand odrl:mediaType ;
                odrl:operator odrl:isAnyOf ;
                odrl:rightOperand ("journals"@en "books"@en) ] ]
    ] ;
    odrl:prohibition [
        rdfs:label "F" ;
        a odrl:Prohibition;
        odrl:action [
            rdf:value cc:Distribution ;
```

```
            odrl:refinement [
                odrl:leftOperand odrl:mediaType ;
                odrl:operator odrl:isAnyOf ;
                odrl:rightOperand ("TV"@en) ] ] ;
        odrl:constraint [
            odrl:leftOperand odrl:dateTime ;
            odrl:operator odrl:lteq ;
            odrl:rightOperand "2040-12-31"^^xsd:date ]
    ] .
```

Listing 3: License L3 expressed with ODRL.

```
:L3 a odrl:Offer ;
    odrl:permission [
        rdfs:label "A.D'" ;
        a odrl:Permission;
        odrl:action [
            rdf:value cc:Distribution ;
            odrl:refinement [
                odrl:leftOperand odrl:mediaType ;
                odrl:operator odrl:isAnyOf ;
                odrl:rightOperand ("online streaming"@en) ] ] ;
        odrl:constraint [
            a odrl:LogicalConstraint ;
            odrl:and ([
                    odrl:leftOperand odrl:dateTime ;
                    odrl:operator odrl:gt ;
                    odrl:rightOperand "2030-12-31"^^xsd:date
                ] [
                    odrl:leftOperand odrl:dateTime ;
                    odrl:operator odrl:lteq ;
                    odrl:rightOperand "2050-12-31"^^xsd:date ] ) ]
    ] , [
        rdfs:label "A.D''" ;
        a odrl:Permission;
        odrl:action [
            rdf:value odrl:compensate ;
            odrl:refinement [
                odrl:leftOperand odrl:mediaType ;
                odrl:operator odrl:isAnyOf ;
                odrl:rightOperand ("TV"@en) ] ] ;
        odrl:constraint [
            a odrl:LogicalConstraint ;
            odrl:and ([
                    odrl:leftOperand odrl:dateTime ;
                    odrl:operator odrl:gt ;
                    odrl:rightOperand "2040-12-31"^^xsd:date
                ] [
                    odrl:leftOperand odrl:dateTime ;
                    odrl:operator odrl:lteq ;
                    odrl:rightOperand "2050-12-31"^^xsd:date  ]  ) ]
    ] ;
    odrl:obligation [
        rdf:label "B" ;
        a odrl:Duty ;
        odrl:action [
            rdf:value odrl:compensate ;
            odrl:refinement [
                odrl:leftOperand odrl:payAmount ;
                odrl:operator odrl:eq ;
                odrl:rightOperand "1000"^^xsd:decimal ;
                odrl:unit <https://dbpedia.org/page/United_States_dollar> ] ]
    ] ;
    odrl:prohibition [
        rdfs:label "C" ;
        a odrl:Prohibition;
        odrl:action cc:CommercialUse ;
        odrl:constraint [
            a odrl:LogicalConstraint ;
            odrl:and ([
                    odrl:leftOperand odrl:dateTime ;
                    odrl:operator odrl:lteq ;
                    odrl:rightOperand "2050-12-31"^^xsd:date
                ] [
                    odrl:leftOperand odrl:spatial ;
                    odrl:operator odrl:isPartOf ;
                    odrl:rightOperand <https://dbpedia.org/page/Europe> ] ) ]
    ] , [
        rdfs:label "F" ;
        a odrl:Prohibition;
        odrl:action [
            rdf:value cc:Distribution ;
            odrl:refinement [
                odrl:leftOperand odrl:mediaType ;
                odrl:operator odrl:isAnyOf ;
                odrl:rightOperand ("TV"@en) ] ] ;
        odrl:constraint [
            odrl:leftOperand odrl:dateTime ;
            odrl:operator odrl:lteq ;
            odrl:rightOperand "2040-12-31"^^xsd:date ]
    ] .
```