www.jusletter-it.eu

Enrico Francesconi

Decidable Reasoning on Provisions and Norms for Legal Information Retrieval and Legal Compliance

The development of legal reasoning using decidable fragments of Semantic Web languages is essential to guarantee the computational tractability in the in the Linked Open Data cloud. This paper introduces a framework for legal knowledge representation and reasoning based on the distinction between the concepts of provision and norm, suited for different kinds of legal reasoning: legal provisions retrieval and norm compliance, respectively. The proposed framework allows the addressed types of reasoning to be implemented using OWL 2 decidable profiles and reasoners. Examples of decidable reasoning within the proposed framework are presented and tested.

Category: Articles
Region: Italy
Field of law: Legal Informatics; Cloud-Computing; Big Data, Open Data & Open Government; Advanced Legal Informatics Systems and Applications; IT-Compliance

Citation: Enrico Francesconi, Decidable Reasoning on Provisions and Norms for Legal Information Retrieval and Legal Compliance, in: Jusletter IT 26 September 2019

Contents

1. Introduction

[1] The transformation of legal regulations in machine readable, actionable rules represents a precondition for developing systems endowed with automatic reasoning facilities for advanced information services in the legal domain.

[2] In literature several approaches have been proposed aiming to formalize legal rules able to support automatic reasoning, as reasoning on deontic notions[1], reasoning for norm compliance[2] or for legal argumentation[3]. When implemented in the Semantic Web, legal reasoning approaches usually utilize languages like OWL/RDF(S) for modeling real world scenarios, as well as mainly SWRL or RIF for representing legal rules for such scenarios. Recently the LegalRuleML standard for legal rules modeling, including rules defeasibility representation and defeasible reasoning, has been proposed[4]. Approaches for modeling real world scenarios and rules often result in non-decidable profiles, so that reasoner are not available or the available ones are not guaranteed to be tractable from a computational point of view, like for LegalRuleML. The current successful trend of Semantic Web implementation according to a Linked Open Data (LOD) approach has produced, and is supposed to produce in the next few years, a huge and growing amount of RDF triples, representing concepts, legal rules and facts. Considering that LOD principles recommend the use of OWL/RDF(S) standards, the availability of decidable reasoners dealing with OWL/RDF(S) modeling is therefore essential, so to guarantee the computational tractability of reasoning in the LOD cloud. Hence the need to represent the semantics of LOD triples by decidable fragments of knowledge modeling languages.

[1] J. M. Broersen, C. Condoravdi, N. Shyam, and G. Pigozzi (Eds.). 2018. *Deontic Logic and Normative Systems – 14th International Conference, DEON 2018*. College Publications, Utrecht, The Netherlands.

[2] R. Muthuri, G. Boella, J. Hulstijn, S. Capecchi, and L. Humphreys. 2017. Compliance patterns: harnessing value modeling and legal interpretation to manage regulatory conversations. In *Proceedings of the 16th International Conference on Artificial Intelligence and Law, ICAIL 2017*. ACM, London, United Kingdom 139–148.

[3] H. Prakken and G. Sartor. 2015. Law and logic: A review from an argumentation perspective. *Artificial Intelligence* 227 (2015), 214–245.

[4] T. Athan, G. Governatori, M. Palmirani, A. Paschke, and A. Wyner. 2015. LegalRuleML: Design Principles and Foundations. In The 11th Reasoning Web Summer School;

G. Governatori, M. Hashmi, H. Lam, S. Villata, and M. Palmirani. 2016. Semantic Business Process Regulatory Compliance Checking Using LegalRuleML. In Knowledge Engineering and Knowledge Management (LNAI), E. Blomqvist, P. Ciancarini, F. Poggi, and F. Vitali (Eds.). Springer International, 746–761.

[3] Also for these reasons in (Gandon et al., 2017)[5] a study has been carried out about how the Linked Open Data framework (including RDF(S)/OWL, LegalRuleML and SPARQL) can be applied to the formalization, publication and processing of legal knowledge, in particular normative requirements and rules.

[4] In terms of knowledge modeling languages for the Semantic Web, OWL DL represents a decidable profile (subset) of OWL, and OWL 2 extends the OWL expressiveness introducing three additional decidable sub-profiles:

- OWL 2 DL: direct extension of OWL DL within the OWL 2 semantics, thus representing a decidable profile of the First Order Logic for OWL 2;
- OWL 2 EL: particularly useful in applications employing ontologies that contain very large numbers of properties and/or classes. It introduces specific restrictions allowing for example existential quantifications, while universal quantification or cardinality restrictions are not allowed;
- OWL 2 QL: particularly useful in applications dealing with a large amount of data. In this profile for example existential quantification to a class expression or a data range are not allowed;
- OWL 2 RL: particularly useful in scalable reasoning without sacrificing too much expressive power. It supports all axioms of OWL 2 apart from disjoint unions of classes (DisjointUnion) and reflexive object property axioms (ReflexiveObjectProperty).

[5] In this paper we introduce a legal reasoning framework based on the distinction between the concepts of Provision and Norm, suited for different kinds of legal reasoning: legal provisions accessibility and norm compliance, respectively. Moreover, an approach based on decidable OWL 2 profiles is presented and tested. The claim of this study is not to address the whole complexity of legal reasoning, including for example non-monotonic reasoning, resolution of norm conflicts[6] or reasoning with incomplete and contradictory information, for which reasoners exist[7]. This study rather aims to present an approach which can be effectively implemented in a decidable framework for the type of reasoning mentioned (provisions retrieval and norms compliance). On the other hand this study may create the ground for investigating how more complex legal reasoning types actually affects the computational burden of the present approach.

[6] This paper is organized as follows: in Section 2 a review of related work about legal reasoning is given, including examples within a description logic computational complexity; in Section 3 the distinction between the concepts of Provision and Norm from the legal theory point of view is discussed[8]; in Section 4 an approach for modeling deontic notions and implementing Hohfeldian

[5] F. Gandon, G. Governatori, and S. Villata. 2017. Normative Requirements as Linked Data. In *Legal Knowledge and Information Systems – Proceeding of the JURIX Conference*, A. Wyner and G. Casini (Eds.), Vol. 302. IOS Press, 1–10.

[6] S. Batsakis, G. Baryannis, G. Governatori, I. Tachmazidis, and G. Antoniou. 2018. Legal Representation and Reasoning in Practice: A Critical Comparison. In *Legal Knowledge and Information Systems – Proceeding of the JURIX Conference*, M. Palmirani (Ed.). IOS Press, 31–40.

[7] Grigoris Antoniou, Nikos Dimaresis, and Guido Governatori. 2008. A system for modal and deontic defeasible reasoning. In *Proceedings of the 2008 ACM Symposium on Applied Computing*. ACM, 2261–2265. https://doi.org/10.1145/1363686.1364226; Ho-Pun Lam and Guido Governatori. 2009. The Making of SPINdle. In *Rule Representation, Interchange and Reasoning on the Web*. Springer. https://doi.org/10.1007/978-3-642-04985-9_29 (all websites last accessed on 27 August 2019).

[8] A. Marmor. 2014. *The Language of Law*. Number 978-0-19-871453-8. Oxford University Press.

reasoning for legal provisions retrieval within a decidable computational framework is recalled[9] and tested on examples; in Section 5 an approach for modeling norms using ontologies, able to implement norm compliance checking within a decidable computational framework is described and tested on examples; in Section 6 some conclusions are reported.

2. Related Work

[7] OWL is the state-of-the-art standard for knowledge modeling in the Semantic Web, effectively used for creating ontologies able to represent concepts and relations of real world scenarios[10]: examples in the legal informatics literature are LRI-Core[11], LKIF[12], CLO[13], DALOS[14], PrOnto[15]. On the other hand legal rules in the Semantic Web have been represented in literature using a variety of languages. (Hoekstra et al., 2009)[16] proposed to use SWRL or RIF in combination with an ontological representation of norms and facts; (Gordon, 2011)[17] introduced rules description using specific XML schemas in combination with ontologies. More recently LegalRuleML[18] as specialization of the RuleML standard has been proposed as language for representing legal rules.

[8] In the last few years several studies have been made to approach the problem of legal reasoning in a decidable computational complexity profile. (Van de Ven et al., 2008)[19] proposed HARNESS, a knowledge-based system developed within the ESTRELLA project, able to implement reasoning on norms for legal assessment, basically the evaluation whether a case is allowed or disallowed given an appropriate body of legal norms. HARNESS includes two knowledge bases: a domain ontology and a set of norms representing the normative articles. Both are de-

[9] E. Francesconi. 2014. A Description Logic Framework for Advanced Accessing and Reasoning over Normative Provisions. *International Journal on Artificial Intelligence and Law* 22, 3 (2014), 291–311; E. Francesconi. 2016. Semantic Model for Legal Resources: Annotation and Reasoning over Normative Provisions. *Semantic Web journal: Special Issue on Semantic Web for the legal domain* 7, 3 (2016), 255–265.

[10] N. Casellas. 2008. *Modelling Legal Knowledge through Ontologies. OPJK: the Ontology of Professional Judicial Knowledge*. Ph.D. Dissertation. Institute of Law and Technology, Autonomous University of Barcelona.

[11] J. Breuker. 2004. Constructing a legal core ontology: LRI-core. In *Proceedings of the Workshop on Ontologies and their Applications*. Porto Alegre, Brazil.

[12] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer. 2007. The LKIF Core Ontology of Basic Legal Concepts. In *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques*, P. Casanovas, M.A. Biasiotti, E. Francesconi, and M.T. Sagri (Eds.). CEUR.org, 43–63. ISSN 1613-0073, CEUR Workshop Proceedings, ISSN 1613-0073, online http://CEUR-WS.org/Vol-321.

[13] A. Gangemi, M.T. Sagri, and D. Tiscornia. 2005. A Constructive Framework for Legal Ontologies. In *Law and the Semantic Web*, Benjamins, Casanovas, Breuker, and Gangemi (Eds.). Springer Verlag.

[14] T. Agnoloni, L. Bacci, E. Francesconi, P. Spinosa, D. Tiscornia, S. Montemagni, and G. Venturi. 2007. Building an ontological support for multilingual legislative drafting. In *Proceedings of the Jurix Conference*. 9–18.

[15] M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, and L. Robaldo. 2018. PrOnto: Privacy Ontology for Legal Reasoning. In *Electronic Government and the Information Systems Perspective. EGOVIS 2018. (Lecture Notes in Computer Science)*, A. Kö, E. Francesconi (Eds.), Vol. 11032. Springer, Cham, 139–152.

[16] R. Hoekstra, J. Breuker, M. di Bello, and A. Boer. 2009. Lkif core: Principled ontology development for the legal domain. In *Law, Ontologies and the Semantic Web*, J. Breuker, P. Casanovas, M. Klein, and E. Francesconi (Eds.). Frontiers in Artificial Intelligence and Applications, Vol. 188. IOS Press, The Netherlands, 21–52.

[17] T. Gordon. 2011. Combining Rules and Ontologies with Carneades. In *Proceedings of the 5th International RuleML2011@BRF Challenge*, Vol. 799. CEUR-WS.org.

[18] OASIS. 2017.LegalRuleML Core Specification Version 1.0. http://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/csprd02/legalruleml-core-spec-v1.0-csprd02.html.

[19] S. Van de Ven, J. Breuker, R. Hoekstra, and L. Wortel. 2008. Automated Legal Assessment in OWL 2. In *Legal Knowledge and Information Systems – Proceeding of the JURIX Conference (Frontiers in Artificial Intelligence and Applications)*, E. Francesconi, G. Sartor, and D. Tiscornia (Eds.), Vol. 189. IOS Press, 170–175.

veloped within the OWL 2 DL profile. Assessment consists of classifying the individuals and properties making up a case description in terms of both ontology and norms simultaneously. The result tells whether norms are violated or not.

[9] As anticipated in Section 1, recently in (Gandon et al., 2017)[20] an approach to represent legal rules as Linked Open Data has been proposed. It aims to respond to the requirements of representing and reasoning on the deontic aspects of normative rules with standard Semantic Web languages, as RDFS and OWL for knowledge representation and SPARQL for inquiries. The rationale of the proposed approach is the coupling of OWL reasoning with SPARQL rules to formalize and implement reasoning (as for example deontic reasoning). In particular normative requirements are represented using LegalRuleML and the deontic conclusions of the legal rules are added to each named graph of the concerned state of affairs.

[10] In (Ceci and Gangemi, 2016)[21] an OWL 2 judicial ontology library (JudO) representing the interpretations performed by a judge when conducting legal reasoning towards the adjudication of a case is illustrated. On the other hand (Ceci, 2013)[22] combines the features of description logic-based ontologies with non-monotonic logics such as defeasible logics.

[11] In this paper we present an approach based on the distinction between Provisions and Norms represented by decidable fragments of OWL 2/RDFS for knowledge modeling and rules representation. This allows us to rely on available decidable reasoners able to derive implicit knowledge and conclusions on the model and related individuals, while leaving to SPARQL the sole task to query the dataset of inferred triples in order to verify deontic conclusions or norm compliance.

3. Provisions and Norms

[12] According to the legal theory point of view, the legal order can be seen as a legal discourse composed by linguistic entities or speech acts[23] with descriptive or prescriptive functions. Every linguistic entity can be seen in a twofold perspective: as a set of signs organized in words and sentences, as well as the meaning of such signs. Following the same twofold view for the legal domain, we can distinguish two levels of interpretation of a linguistic entity expressing a legal rule: in terms of a set of signs organized in words and sentences for creating a normative

[20] F. Gandon, G. Governatori, and S. Villata. 2017. Normative Requirements as Linked Data. In *Legal Knowledge and Information Systems – Proceeding of the JURIX Conference*, A. Wyner and G. Casini (Eds.), Vol. 302. IOS Press, 1–10.

[21] M. Ceci and A. Gangemi. 2016. An OWL ontology library representing judicial interpretations. *Semantic Web Journal – Special Issue on Semantic Web for the Legal Domain* 7, 3 (2016), 229–253.

[22] M. Ceci. 2013. Representing Judicial Argumentation in the Semantic Web. In *Proceedings of the V Workshop on Artificial Intelligence and the Complexity of Legal Systems (AICOL)*, P. Casanovas, U. Pagallo, M. Palmirani, and G. Sartor (Eds.). Springer, 172–187.

[23] J.R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Number ISBN 978-0521096263. Cambridge University Press.

statement, typically called Provision[24], and in terms of the meaning of such normative statement, typically called Norm[25].

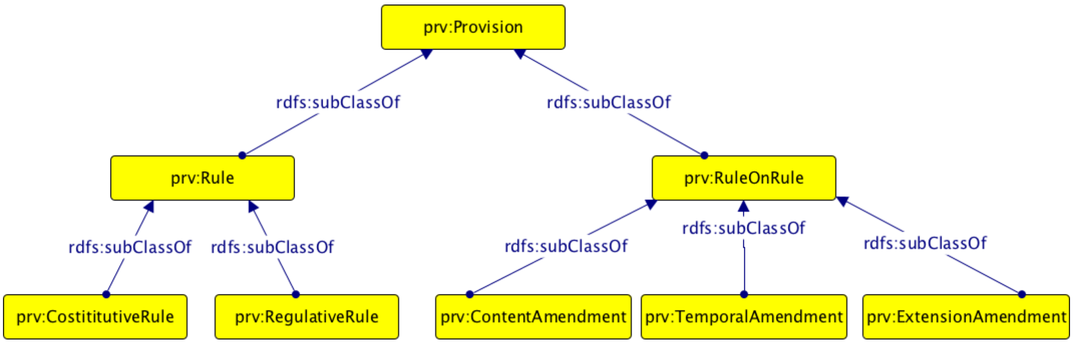


Figura 1: The Provision Model top classes («prv» is the namespace of provisions)

[13] Provisions have been classified in (Biagioli, 2009)[26] in terms of provision types, organised into two main groups (Fig. 1): Rules and Rules on Rules. Rules can be Constitutive Rules as Definition introducing entities, or Regulative Rules as the deontic concepts Duty, Right, Power, etc., regulating subject roles and activities. Rules on Rules are different kinds of amendments: Temporal, Extension or Content amendments as Insertion, Substitution, Repeal. Each provision type is characterized by specific attributes (for example the Bearer or the Counterpart of a Right), reflecting the lawmaker directions. Provision types and attributes can be considered as a sort of metadata model able to analytically describe fragments of legislative texts, hence the name of Provision Model[27]. In Fig. 2 the Provision Model classes prv:Right and prv:Duty, and the related properties (as prv:hasRightBearer, prv:hasDutyCounterpart, etc.), under the namespace prv: for provisions in the Semantic Web, are sketched.

[24] C. Biagioli. 2009. *Modelli Funzionali delle Leggi. Verso testi legislativi autoesplicativi.*Legal Information and Communications Technologies Series, Vol. 6. European Press Academic Publishing, Florence, Italy; J. Raz. 1980. *The Concept of a Legal System*. Oxford University Press.

[25] P. Ciancarini, F. Poggi, and F. Vitali (Eds.). Springer International, 746–761. R. Guastini. 2010. *Le Fonti del Diritto. Fondamenti teorici.* Giuffrè, Milano; A. Marmor. 2014. *The Language of Law*. Number 978-0-19-871453-8. Oxford University Press.

[26] C. Biagioli. 2009. *Modelli Funzionali delle Leggi. Verso testi legislativi autoesplicativi.*Legal Information and Communications Technologies Series, Vol. 6. European Press Academic Publishing, Florence, Italy.

[27] C. Biagioli. 1996. Law Making Environment: model based system for the formulation, research and diagnosis of legislation. (Retrieved on 20 January 2019).

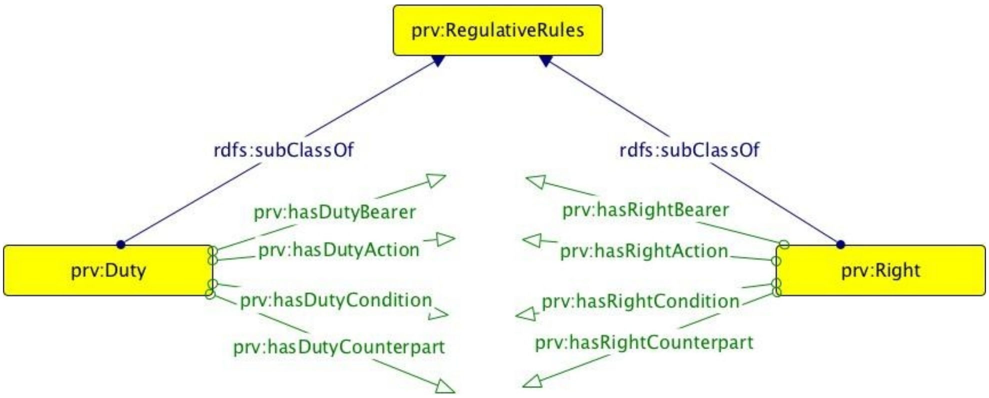


Figura 2: Examples of provisions attributes, represented as ontology classes and properties.

[14] Norms represent the way how provisions are applied; as such they represent the product of an interpretative process[28].

[15] Provisions and related norms have different roles and properties pertaining to the different abstraction levels they operate at. The need of distinguishing between provisions and norms becomes essential when we observe that there may be not a bijective relationship between them: a norm can be expressed by different provisions, as well as it can be valid the opposite, namely one provision can include more norms[29]. Moreover, they have different relationships with time. Provisions, as pure textual objects, are the product of lawmaking (legal drafting activity and promulgation) and they have a specific relation with time represented by the in-force date, namely the starting date of their existence in the legal order. On the other hand norms are the meaning of provisions, namely their applicative interpretation; as such they have a specific relation with time represented by the efficacy date, namely the starting date in which a norm can be concretely applied. Therefore, while it is obvious that we can have cases of provisions not in-force and related norms not effective, as well as provisions in-force and related norms effective, we can also have provisions in-force and related norms not effective, as well as the symmetric case (this last one is usually referred as ultra-activity of a norm).

[16] Having different nature, such concepts operate in different domains. A provision, as pure textual object, represents the building block of the legal order (new provisions can enter or leave the legal order). On the other hand norms, can either modify the text of other provisions (in case of different type of amendments) or can introduce restrictions on the real world (in case of obligations, for example).

[17] We have underlined the different nature of such objects as a background for introducing an approach for legal reasoning using such different concepts in different types of legal reasoning. Advanced legal document retrieval systems able, for example, to implement Hohfeldian reasoning on deontic notions, is a type of reasoning managing textual information, thus pertaining to provisions. Legal compliance checking is a process aiming to verify if a fact, occurring in the real world, complies with a legal norm. Real world scenarios and facts can be effectively represented in terms of ontologies and related individuals, respectively. Norms, which facts have to be compliant with, provide constraints on the reality, therefore they can be modeled, in particular as

[28] H. Kelsen. 1991. *General Theory of Norms*. Clarendon Press, Oxford.

[29] G. Pino. 2016. *Teoria analitica del diritto*. Number 9788846744517. ETS, Chapter2. Norma giuridica, 144–183.

far as obligations are concerned, as restrictions on ontology properties, used for legal compliance checking.

[18] Let's consider two examples of rules, R1 and R2, to illustrate our approach:

- R1 : *The supplier shall communicate to the consumer all the contractual terms and conditions*
- R2 : *According to a [country] law one cannot drive over 90 km/h*

[19] Both rules are speech acts, namely Provisions in specific regulations. In this sense R1 can be classified as a Duty of a Supplier towards Consumer, while R2 can be classified as an Obligation for any Driver in the related specific country. Such classification can be used for implementing an advanced retrieval system able to select provisions by provision types and attributes. Moreover, R1 is an example in which Hohfeldian reasoning can be applied, since that provision can be seen as a Right of the Consumer to receive communications by Supplier. Such provision has to be retrieved either if we search for the supplier's duties or if we search for the consumer's rights. In all these cases an advanced legal retrieval system endowed with legal reasoning facilities pertains to provisions.

[20] When we consider the application of R1 and R2 on specific facts, we actually talk about Norms. As previously discussed, norms which facts have to be compliant with, can be viewed as constraints on the real world to be regulated.

[21] In the case of R1, the related scenario can be modeled in terms of an ontology including a class Supplier, having a boolean property hasCommunicatedConditions, while the norm R1 can be modeled in terms of restriction on that property, so that the compliant supplier must have set to «true» the value of such property when he has complied with his communication duties.

[22] Similarly, in case of R2, the vehicles circulation scenario of a specific country can be modeled in terms of an ontology including a class Driver, having a property hasDrivingSpeed, while the norm R2 can be modeled in terms of restriction on that property, so that the compliant individual drivers must have set the value of such property in the interval [0.0 Km/h, 90.0 Km/h] (detected for example by the police through a speed checker station). In both cases legal reasoning in terms of legal compliance checking can be obtained by managing norms modeled as restrictions on property values of ontology classes.

[23] In Section 4 we recall the modeling approach, described in (Francesconi 2014 and 2016)[30], about how legal reasoning on deontic notions (as reasoning over Hohfeldian relations) for an advanced legal provisions retrieval system can be modeled within description logic implemented in OWL 2 DL. Similarly, in Section 5 we show how legal reasoning for norm compliance can be implemented by modeling norms as ontology properties restriction using decidable fragments of OWL 2. In both cases inferences and related legal reasoning can be effectively calculated using decidable reasoners.

[30] E. Francesconi. 2014. A Description Logic Framework for Advanced Accessing and Reasoning over Normative Provisions. *International Journal on Artificial Intelligence and Law* 22, 3 (2014), 291–311; E. Francesconi. 2016. Semantic Model for Legal Resources: Annotation and Reasoning over Normative Provisions. *Semantic Web journal: Special Issue on Semantic Web for the legal domain* 7, 3 (2016), 255–265.

4. Modeling Provisions for advanced legal Provisions Accessibility

[24] In (Francesconi, 2014 and 2016)[31] it is shown how Hohfeldian relations on deontic and potestative notions can be managed within a description logic computational framework. We recall here the main aspects of the approach to show, for the examples R1 and R2, how Provisions can be used to implement an advanced legal provisions retrieval system, endowed with legal reasoning facilities, using a decidable fragment of OWL 2 (in particular OWL 2 DL), therefore exploiting existing decidable reasoners.

[25] In this recall, we show the approach for deontic notions and their relations, sketched in the schema of Fig. 3.[32]

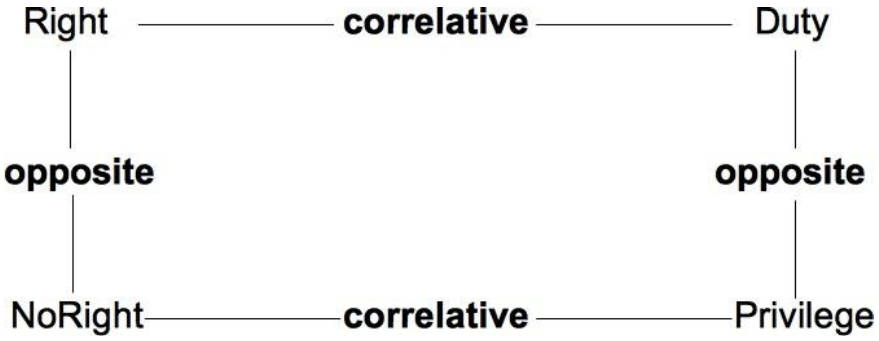


Figura 3: Hohfeldian relations on deontic concepts.

[26] In order to implement an advanced legal provisions retrieval system, it is necessary to describe the relations between provisions at the level of the Provision Model. For example the Hohfeldian relation between Duty and Right can be effectively represented by observing that a Right, in correlative correspondence with a Duty, is actually not explicitly expressed in the text, but represents an implicit provision, basically a different view of the Duty itself, where the values of the related bearer and counterpart attributes are swapped. Therefore, the Provision Model can be extended in terms of Duty and Right[33] implicit and explicit disjoint subclasses, able to represent a complete covering of the related superclass (ex: ExplicitRight and ImplicitRight disjoint subclasses represent a complete covering of the Right superclass).

[27] Attributes can also be specified as regards both implicit and explicit provisions, so that hasImplicitDutyBearer and hasExplicitDutyBearer are sub-properties of hasDutyBearer, as well as hasImplicitRightBearer and hasExplicitRightBearer are sub-properties of hasRightBearer.

[28] To represent the Hohfeldian fundamental relations between Duty and Right, firstly an equivalence relation between their explicit and implicit views is established: ImplicitRight ≡ ExplicitDuty and ImplicitDuty ≡ ExplicitRight. In Fig. 4 the established sub-class and equivalence relations between Duty and Right in their explicit and implicit views are summed up.

[31] E. Francesconi. 2014. A Description Logic Framework for Advanced Accessing and Reasoning over Normative Provisions. *International Journal on Artificial Intelligence and Law* 22, 3 (2014), 291–311; E. Francesconi. 2016. Semantic Model for Legal Resources: Annotation and Reasoning over Normative Provisions. *Semantic Web journal: Special Issue on Semantic Web for the legal domain* 7, 3 (2016), 255–265.

[32] More details on this modeling approach and its application to potestative notions (*Power, Liability, Disability Immunity*), can be found in (Francesconi, 2014 and 2016, Fn. 31.).

[33] Where «prv:», namespace for provisions, is hereinafter implied.

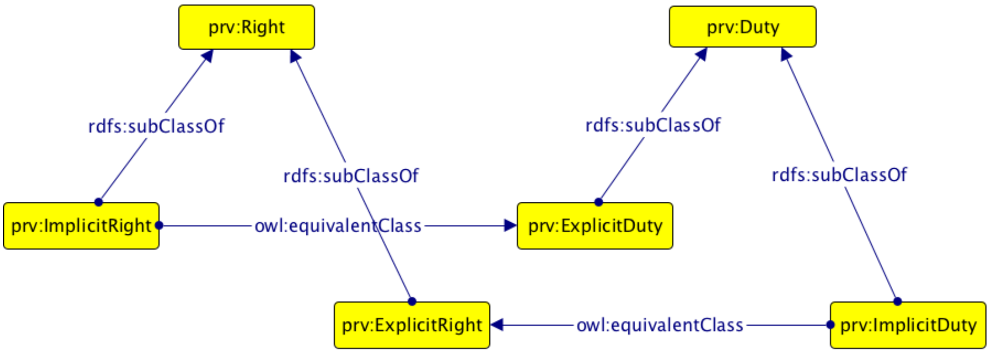


Figura 4: Sub-class and asserted equivalence relations between Duty/Right deontic correlative provisions.

[29] Moreover, equivalence relations between implicit/explicit Duty and Right attributes can be established. In Fig. 5 the asserted properties of ExplicitDuty and ImplicitRight and their mutual equivalence relations are shown (hasImplicitRightBearer ≡ hasExplicitDutyCounterpart and hasImplicitRightCounterpart ≡ hasExplicitDutyBearer).

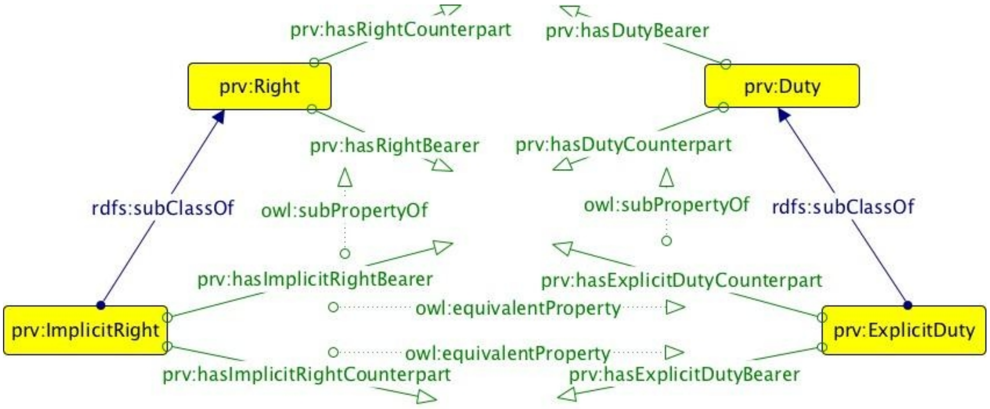


Figura 5: Asserted properties of ExplicitDuty and ImplicitRight and their mutual equivalence relations.

[30] The same holds for the asserted properties of ImplicitDuty and ExplicitRight and their mutual equivalence relations (hasImplicitDutyBearer ≡ hasExplicitRightCounterpart and hasImplicitDutyCounterpart ≡ hasExplicitRightBearer) (Fig. 6)

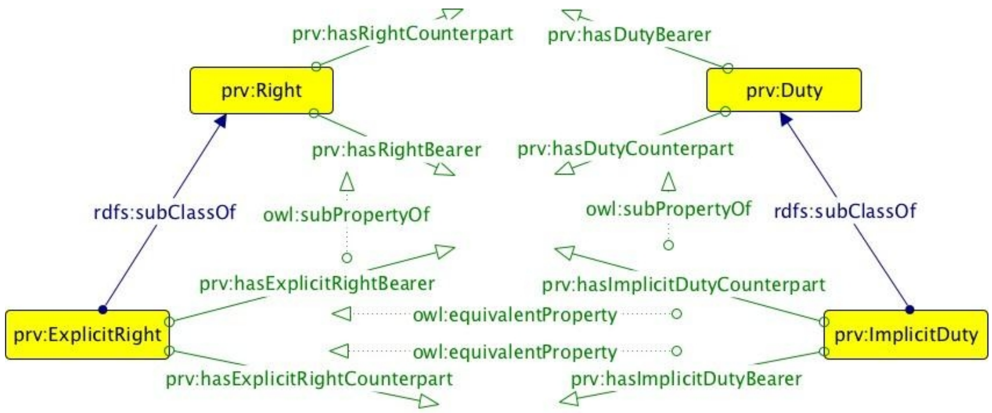
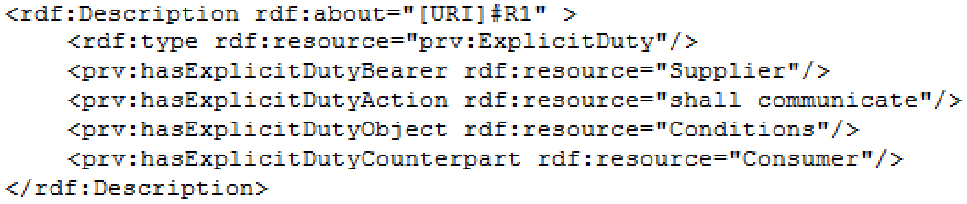


Figura 6: Asserted properties of ImplicitDuty and ExplicitRight and their mutual equivalence relations.

[31] Note that the proposed patterns do not interfere with the relations between Right and Duty, which still hold. In fact, for the couple Right/Duty, an individual of ExplicitDuty is also an individual of Duty, given the axiom rdfs:subClassOf(ExplicitDuty, Duty). Moreover the axiom owl:equivalentClass(ImplicitRight, ExplicitDuty) tells us that such individual is also an ImplicitRight, which is also a Right, given the axiom rdfs:subClassOf(ImplicitRight, Right). Since this is done symmetrically for explicit and implicit duties and rights, we can deduce that Right is equivalent to Duty, namely is another reading of the Duty itself, given that the union of the disjoint explicit and implicit subclasses covers completely the related superclass.

4.1. Provisions R1 and R2 representation and reasoning

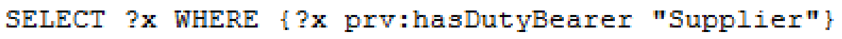
[32] Such modeling approach can be used to represent the provisions R1 and R2 in a way that a provision retrieval system can be implemented using a decidable reasoner. For example R1 can represented as follows:


```
<rdf:Description rdf:about="[URI]#R1" >
    <rdf:type rdf:resource="prv:ExplicitDuty"/>
    <prv:hasExplicitDutyBearer rdf:resource="Supplier"/>
    <prv:hasExplicitDutyAction rdf:resource="shall communicate"/>
    <prv:hasExplicitDutyObject rdf:resource="Conditions"/>
    <prv:hasExplicitDutyCounterpart rdf:resource="Consumer"/>
</rdf:Description>
```

[33] where the values of the provisions attributes are literals. Note that only explicit provision classes (and consequently explicit properties) are used to annotate textual provisions, as they are the only provisions actually (explicitly) expressed in the text, while implicit provisions act as a sort of «abstract» classes, which are used for reasoning.

[34] As both the Provision Model (and related instances) result in OWL 2 DL profile, inferences can be calculated through an OWL 2 DL reasoner. In this example the Pellet[34] Java based OWL 2 DL reasoner is used to derive the inferred model.

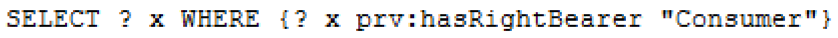
[35] Let's assume to query the system as follows in order to retrieve the suppliers' duties:



```
SELECT ?x WHERE {?x prv:hasDutyBearer "Supplier"}
```

where x is the variable that will contain the identifiers of the retrieved provisions instances. In case the non-inferred model is queried, no provisions are retrieved since only ExplicitDuty and related attributes are used for provision annotation. In case the inferred model is queried, the inferred provisions are retrieved, either annotated as ExplicitDuty of Supplier, if any, or implicitly deduced by provision relations. By exploiting the established rdfs:subClass relations between provisions type and attributes, the system will act as virtually expanding the query in terms of hasExplicitDutyBearer and hasImplicitDutyBearer, thus being able to retrieve R1, which is the provision having Supplier as value of the property hasExplicitDutyBearer.

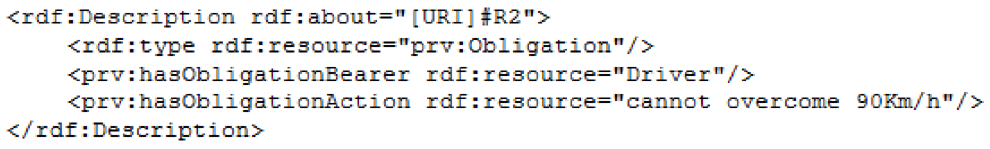
[36] Similar considerations can be given if we want to retrieve the consumer' rights, as follows:



```
SELECT ? x WHERE {? x prv:hasRightBearer "Consumer"}
```

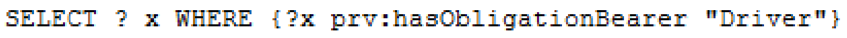
In this case Hohfeldian reasoning is produced. In fact by exploiting the established rdfs:subClass and owl:equivalentClass relations between provisions type and attributes, the system will act as virtually expanding the query in terms of hasExplicitRightBearer and hasImplicitRightBearer. For the last one the following relation holds hasImplicitRightBearer ≡ hasExplicitDutyCounterpart: this allows the system to retrieve R1, which is the provision having Consumer as value of the property prv:hasExplicitDutyCounterpart. Since this is the result of axioms established in the Provision Model for implementing Hohfeldian relations, the result is a Hohfeldian reasoning over provisions (namely searching for consumers' rights and retrieving the related suppliers' duties).

[37] Very similar considerations can be made for the annotation of R2 and a query able to retrieve it. The provision R2 can be represented as follows:



```
<rdf:Description rdf:about="[URI]#R2">
    <rdf:type rdf:resource="prv:Obligation"/>
    <prv:hasObligationBearer rdf:resource="Driver"/>
    <prv:hasObligationAction rdf:resource="cannot overcome 90Km/h"/>
</rdf:Description>
```

where the values of the provisions attributes are literals. The following SPARQL query is able to retrieve the provision R2:



```
SELECT ? x WHERE {?x prv:hasObligationBearer "Driver"}
```

[34] https://www.w3.org/2001/sw/wiki/Pellet (last accessed on 20 August 2019).

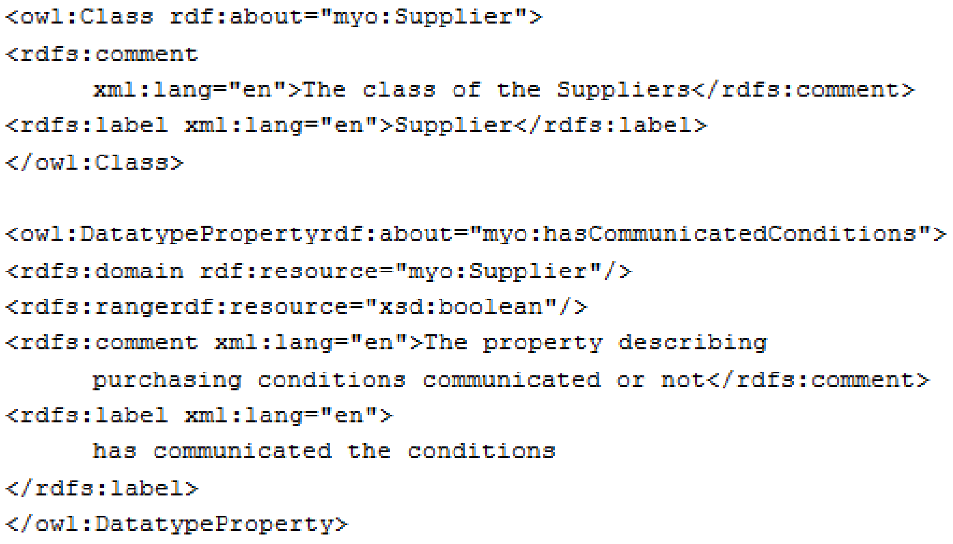
[38] Both the previous cases represent retrieval examples including legal reasoning on deontic notions which can be managed within the OWL 2 DL decidable computational profile.

5. Modeling Norms for legal Compliance Checking

[39] As discussed in Section 3, norms can be viewed as the application, subject to interpretation, of legal provisions, providing constraints on a real world scenario to be regulated. In the Semantic Web a real world scenario is usually represented by a domain ontology. In this context a norm, providing constraints to such scenario, can be modeled in terms of constraints on the domain ontology: for example, in case of obligations, as ontology property restrictions. In the following of this section we show how the norms expressed in R1 and R2, can be represented as restrictions on the addressed scenarios, and how such a representation can be used for norm compliance checking, within a computational complexity decidable profile. Note that in the examples, the relations between text, provisions and corresponding norms have not been reported for the sake of simplicity, but can obviously be expressed by relations between the related URIs.

5.1. Norm R1 representation and facts Compliance Checking

[40] In the case of R1, the scenario can be modeled in terms of an ontology including a class Supplier, having a boolean property hasCommunicatedConditions. In OWL 2 terms the scenario concerning R1 can be expressed as follows:



```
<owl:Class rdf:about="myo:Supplier">
<rdfs:comment
      xml:lang="en">The class of the Suppliers</rdfs:comment>
<rdfs:label xml:lang="en">Supplier</rdfs:label>
</owl:Class>

<owl:DatatypePropertyrdf:about="myo:hasCommunicatedConditions">
<rdfs:domain rdf:resource="myo:Supplier"/>
<rdfs:rangerdf:resource="xsd:boolean"/>
<rdfs:comment xml:lang="en">The property describing
      purchasing conditions communicated or not</rdfs:comment>
<rdfs:label xml:lang="en">
      has communicated the conditions
</rdfs:label>
</owl:DatatypeProperty>
```

where myo: is a fictitious namespace for a fictitious ontology «MyOntology».

[41] Norm R1, expressing a duty for the suppliers states that suppliers must communicate purchasing conditions to the consumers: the individuals of the class Supplier complying with this norm are all those ones belonging to the subclass SupplierR1Compliant identified by a restriction on the boolean property hasCommunicatedConditions to have value «true» (Fig. 7).

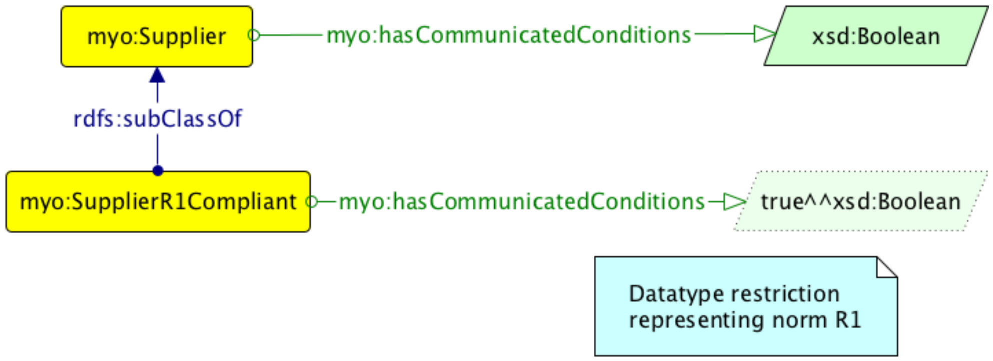
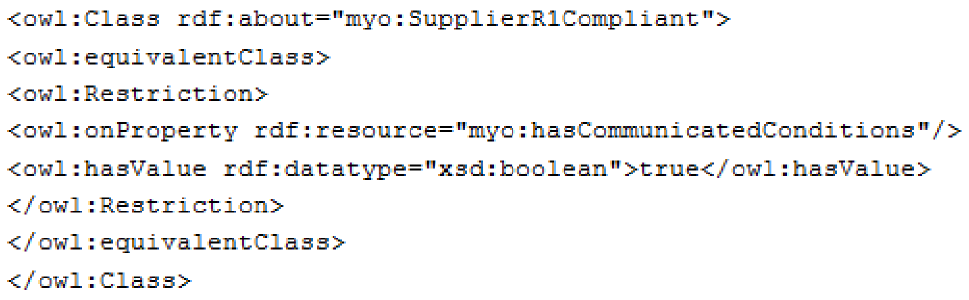

myo:Supplier
myo:hasCommunicatedConditions
xsd:Boolean
rdfs:subClassOf
myo:SupplierR1Compliant
myo:hasCommunicatedConditions
true^^xsd:Boolean
Datatype restriction representing norm R1

Figura 7: Norm R1 represented as restriction on the Supplier's property hasCommunicated-Conditions (note that the subclass relation between SupplierR1Compliant and Supplier is inferred).

[42] In other terms the norm R1 is represented as restriction on the property hasCommunicated-Conditions able to identify the class SupplierR1Compliant which is equivalent (see owl:equivalentClass relation here below) to the class of all individuals for which the value of the property under consideration is «true», as follows:

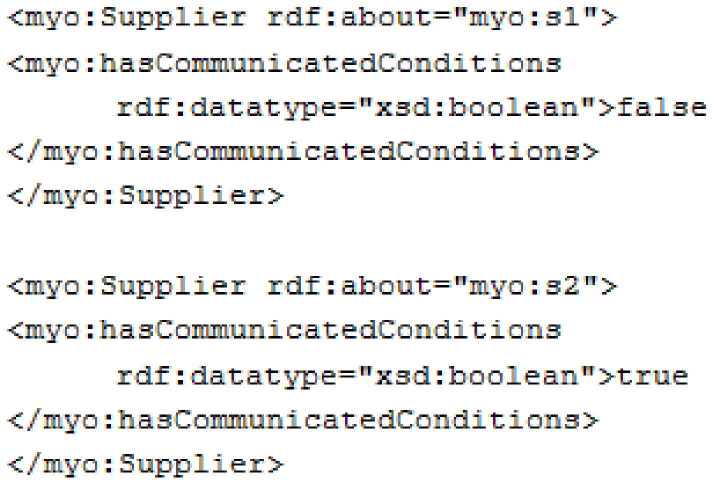


```
<owl:Class rdf:about="myo:SupplierR1Compliant">
<owl:equivalentClass>
<owl:Restriction>
<owl:onProperty rdf:resource="myo:hasCommunicatedConditions"/>
<owl:hasValue rdf:datatype="xsd:boolean">true</owl:hasValue>
</owl:Restriction>
</owl:equivalentClass>
</owl:Class>
```

[43] Such a representation for the real world scenario and related norm expressed by R1 results in the OWL 2 DL, as well as OWL 2 RL, decidable profiles.[35] This allows us to use a OWL 2 DL decidable reasoner, as for example Pellet, in order to implement reasoning facilities, preparing the ground for compliance checking with respect to R1. The inferred model produced by Pellet establishes a rdfs:subClassOf relationship between SupplierR1Compliant and Supplier (as shown in Fig. 7), where SupplierR1Compliant is the class of all the individuals of type Supplier having «true» as value of the property hasCommunicatedConditions. Therefore, compliance checking according to the norm R1 is a problem of checking if an individual of type Supplier belongs also to the class SupplierR1Compliant.

[44] As an example let's consider the following two individuals myo:s1 and myo:s2 of the class Supplier:

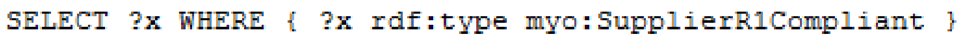
[35] As it can be verified using the Manchester validator at http://mowl-power.cs.man.ac.uk:8080/validator/.



```
<myo:Supplier rdf:about="myo:s1">
<myo:hasCommunicatedConditions
      rdf:datatype="xsd:boolean">false
</myo:hasCommunicatedConditions>
</myo:Supplier>

<myo:Supplier rdf:about="myo:s2">
<myo:hasCommunicatedConditions
      rdf:datatype="xsd:boolean">true
</myo:hasCommunicatedConditions>
</myo:Supplier>
```

myo:s1 is an individual not compliant with R1, while myo:s2 is complaint with R1. The following SPARQL query

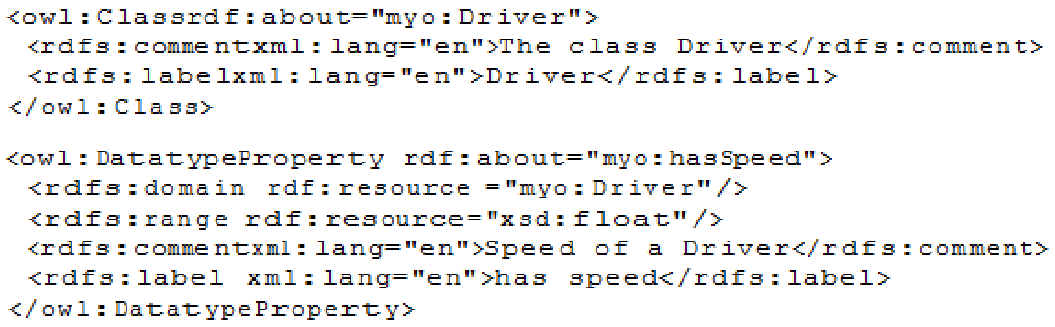


```
SELECT ?x WHERE { ?x rdf:type myo:SupplierR1Compliant }
```

is able to select the individuals which are complaint with R1 (in our case s2). Legal reasoning in terms of norm compliance checking is therefore performed within a decidable computational complexity profile.

5.2. Norm R2 representation and facts Compliance Checking

[45] In the case of R2, the vehicles drivers circulation scenario can be modeled in terms of an ontology including a class Driver, having a datatype property hasDrivingSpeed with range in the xsd:float datatype. In OWL 2 terms, the vehicles drivers circulation scenario concerning R2 can be expressed as follows:



```
<owl:Classrdf:about="myo:Driver">
  <rdfs:commentxml:lang="en">The class Driver</rdfs:comment>
  <rdfs:labelxml:lang="en">Driver</rdfs:label>
</owl:Class>

<owl:DatatypeProperty rdf:about="myo:hasSpeed">
  <rdfs:domain rdf:resource ="myo:Driver"/>
  <rdfs:range rdf:resource="xsd:float"/>
  <rdfs:commentxml:lang="en">Speed of a Driver</rdfs:comment>
  <rdfs:label xml:lang="en">has speed</rdfs:label>
</owl:DatatypeProperty>
```

[46] Norm R2, expressing an obligation on the drivers circulation scenario, states that, according to the related country law, one cannot drive over 90 km/h: the individuals of the class Driver

complying with this norm are those ones belonging to the subclass DriverR2Compliant having value [0.0 km/h, 90.0 Km/h] on the datatype property hasDrivingSpeed (Fig. 8).

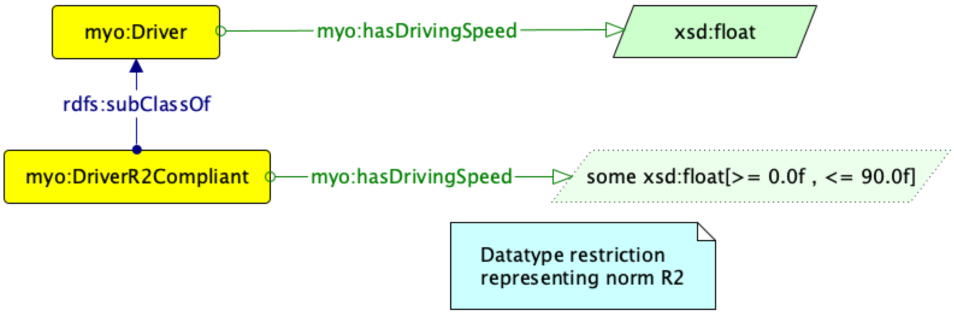


Figura 8: Norm R2 represented as restriction on the Driver's property hasDrivingSpeed (note that the subclass relation between DrivingR2Compliant and Driver is inferred).

[47] In other terms the norm R2 is represented as restriction on the property hasDrivingSpeed able to identify the class DriverR2Compliant which is equivalent to the class of the individuals for which the values of the property under consideration are in the range [0.0 km/h, 90.0 km/h]. In order to represent such constraints the following restriction on the datatype property myo:hasDrivingSpeed to values (inclusively) between 0.0 Km/h and 90.0 km/h can be expressed by the xsd:minInclusive and xsd:maxInclusive datatype bound properties. In OWL 2 this results as follows:

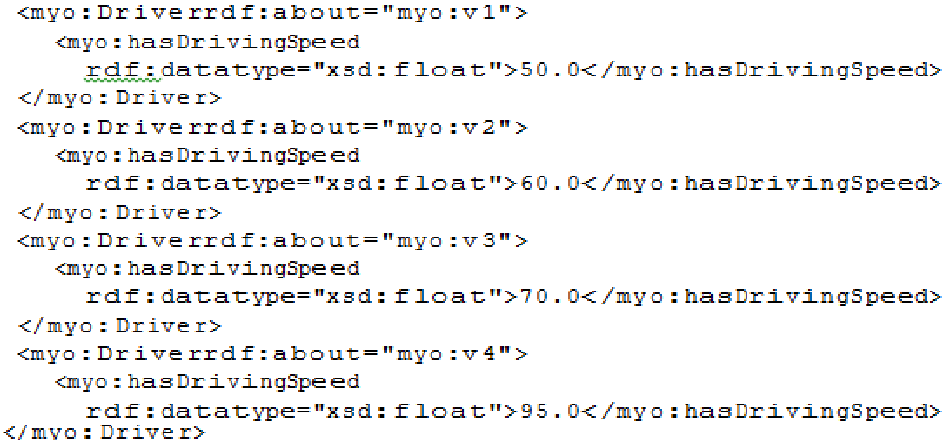
```
 < owl:Class rdf:about ="myo:DriverR2Compliant" >
  < owl:equivalentClass>
  < owl:Restriction>
    < owl:onProperty rdf:resource="myo:hasDriving Speed" />
    <owl:someValuesFrom>
      <rdfs:Datatype>
      <owl:onDatatype  rdf:resource="xsd:float"/>
      <owl:withRestrictions rdf:parseType="Collection">
      <rdf:Description>
        <xsd:minInclusive
          rdf:datatype="xsd:float">0.0</xsd:minInclusive>
      </rdf:Description>
      <rdf:Description>
        <xsd:maxInclusive
         rdf:datatype="xsd:float">90.0</xsd:maxInclusive>
      </rdf:Description>
      </owl:withRestrictions>
      </rdfs:Datatype>
    </owl:someValuesFrom>
  </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

[48] Such a representation results in the OWL 2 DL decidable profile[36]. As in the previous example, the inferred model, produced by Pellet, establishes a rdfs:subClassOf relationship between

[36] As it can be verified using the Manchester validator at http://mowl-power.cs.man.ac.uk:8080/validator/.

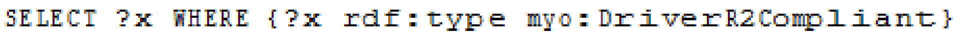
DriverR2Compliant and Driver (as shown in Fig. 8), where DriverR2Compliant is the class of all the individuals of type Driver having values of the property hasDrivingSpeed in the interval [0.0 km/h, 90.0 km/h]. Therefore, compliance checking according to the norm R2 is a problem of checking if an individual of type Driver belongs to the class DriverR2Compliant.

[49] As a concrete example, let's consider the following four individuals of the class Driver:



```
<myo:Driverrdf:about="myo:v1">
   <myo:hasDrivingSpeed
     rdf:datatype="xsd:float">50.0</myo:hasDrivingSpeed>
</myo:Driver>
<myo:Driverrdf:about="myo:v2">
   <myo:hasDrivingSpeed
     rdf:datatype="xsd:float">60.0</myo:hasDrivingSpeed>
</myo:Driver>
<myo:Driverrdf:about="myo:v3">
   <myo:hasDrivingSpeed
     rdf:datatype="xsd:float">70.0</myo:hasDrivingSpeed>
</myo:Driver>
<myo:Driverrdf:about="myo:v4">
   <myo:hasDrivingSpeed
     rdf:datatype="xsd:float">95.0</myo:hasDrivingSpeed>
</myo:Driver>
```

[50] In this list, the individual myo:v4 is not compliant with R2 (having speed 95.0 Km/h ≥ 90.0 Km/h). The following query:



```
SELECT ?x WHERE {?x rdf:type myo:DriverR2Compliant}
```

is able to select the individuals which are complaint with R2 (in our case myo:v1, myo:v2, myo:v3). Also in this case the norm compliance checking is performed within a decidable computational complexity profile.

6. Conclusions and future Developments

[51] In this paper we have presented a legal reasoning approach based on the distinction between the concepts of provisions and norms, able to deal with different types of legal reasoning, in particular advanced legal provisions retrieval, as well as norms compliance checking. The method is based on the use of decidable fragments of OWL 2, able to guarantee the computational tractability of the approach.

[52] Actually, legal reasoning is characterized by a more complex logic, the typical case being the defeasible one. On the other hand, the aim of this approach is to identify a framework, which seems promising in terms of computational tractability, under whose umbrella investigating the sufficient conditions according to which some types of legal reasoning can be managed by decidable reasoners, so to guarantee not only a computational tractability, but also the possibility to reuse existing reasoners developed for decidable profiles of OWL 2.

[53] In the near future we aim to explore the semantics and use of *DL-safe Rules*[37], which are a specific decidable fragment of SWRL, able to describe a type of rules where variables appears in the rule premise in both unary and binary predicates.

Enrico Francesconi, Institute of Legal Informatics and Judicial Systems, Italian National Research Council (IGSG-CNR, Italy) and Publications Office of the European Union (Luxembourg).

[37] P. Hitzler, M. Krötzsch, and S. Rudolph. 2009. *Foundations of the Semantic Web Technologies*. Chapman & Hall/CRC.