Towards a Legal Reasoning System based on Description Logics - A Position Paper -

Makoto HARAGUCHI
Division of Electronics and Information Engineering
Hokkaido University
N-13, W-8, Sapporo 060, JAPAN
Tel & Fax: +81-11-706-7106.
E-mail: makoto@db.huee.hokudai.ac.jp

1 Why description logic for legal reasoning? - A general explanation -

Legal reasoning is concerned with various kinds of legal concepts that relate to another concepts used in our daily life. Although those legal concepts are abstract in the sense that there always exist possibilities to interpret them in various ways, lawyers seem to understand them as real things at least in their "legal reality". For instance, Japanese Civil Codes include a notion of falsity that means a false declaration of intention. In spite of its abstractness, well-trained lawyers can judge that one legal act involves the falsity and that the others do not.

From this point of view, it seems natural to consider that each legal act has its legal identity. Since the standard first order logic (FOL, for short) is founded on the assumption that every object treated within the logic has its identity, it seems natural to develop a way of representing the legal acts as first-order terms denoting individual objects. However, they are highly related each other at their conceptual level. For instance, a person fills the role of agent in a contract. Thus the two concepts "person" and "contract" are related by the role "agent".

Recently in Japan, some researchers [12; 17] try to develop a legal ontology consisting legal concepts and conceptual relationships between them. The latter ones can be roles in description logics. So if we like to build a large legal knowledge base, we would have a description logic containing many legal conceptual terms linked by roles. Based on this, an efficient reasoner (RKB interpreter) would be designed as an extension of FOL prover, for both DL and FOL makes the same assumption that each object has its identity and every concept is a set of objects.

2 Order-sorted logic versus DL

We have briefly discussed why DL is suitable and natural for legal reasoning. Although this is one of the reasons why I am interested in DL, I have to mention my research history which guides me in DL.

Before knowing DL, I have been studying a way of representing and learning legal knowledge in terms of order-sorted logic [6; 18]. In the studies, each legal concept and its instance are encoded as a sort (symbol) and an well-sorted FOL term, respectively. For instance, a sentence

> "a contract that is made by a person a with a person b for a property c is a contract. "

is encoded as a sorted term

$$contract\_f(person\_a, person\_b, property\_c) : contract,$$

where $contract\_f$ is a function symbol with domain sorts $person^2 \times object$ and a codomain sort $contract$. Thus the function symbol works to produce an instance of $contract$ whose identity is determined by its arguments.

Assuming such an order-sorted signature with a sort hierarchy (taxonomic hierarchy), we represent legal rules as sorted clauses, such as

$$\text{legal\_effect}(X : contract) : -\text{legal\_requirements}(X).$$

Furthermore, to make a hypothetical legal rule, we can introduce a notion of sorted generalization that replaces the sorted variable $X : contract$ with a variable Y of more general sort $juristic\_act$. This produces a hypothetical rule

$$\text{legal\_effect}(Y : juristic\_act) : -\text{legal\_requirements}(Y),$$

that has more applicability. Thus the original rule is now generalized and is therefore analogically applicable to similar legal concepts such as "registration" for instance.

The points here we have to keep in our mind are as follows:

1. Although we have developed the analogical application of legal rules according to a similarity our legal taxonomy contains, we can also have the same methodology for legal knowledge represented in DL. So my first intention is to reconstruct a legal reasoning by analogy under the framework of DL.
2. Information about legal notions and legal events is not necessarily completely presented. Occasionally, some of the parties of contracts are not explicitly given. In such a case, we have to introduce Skolem constants tentatively to fill the missing information. This is due to our representational assumption that each legal acts are formed by a function application.

A problem occurs when such a missing information about legal acts appears later, namely by updating our knowledge base. Then we have to reason about the equality between Skolem constant and the term newly obtained. This will increase additional tasks for our reasoner.

DL, on the other hand, can give a name to each legal act by an individual constant whose identity is given. The addition of new information about the act is simply performed by asserting additional relations to RKB.

The larger knowledge base we are handling, the more the second point becomes important. Taking the two points above into account, I am now shifting our underlying language from order-sorted formalism to DL.

3 More about Analogy

Legal taxonomic hierarchy can be viewed as a way of defining legal concepts. Then some similarities between legal concepts are naturally deduced from the structure of hierarchy, assuming an inheritance procedure along the hierarchy. Thus the similarities represented by the taxonomy are completely determined by a way we define the concepts. Consequently, they are said to be definitional. The definitional similarities apart, we can have various similarities not represented in the hierarchy. Such a similarity plays an important role in reasoning about a new case with provisoes ones that are similar with some respects.

Thus, to increase the power of our analogical reasoner, we have to detect similarities not derived from the taxonomic hierarchy. From this viewpoint, we have recently proposed a new framework, called a Goal-Dependent Abstraction (GDA) [13],[8] for doing that based on a notion of theory abstraction [14]. In a word, given an order-sorted representation of legal knowledge and a goal G, our GDA algorithm tries to find a sort mapping $\varphi : Sort_{concrete} \rightarrow Sort_{abstract}$ such that the proof of the goal G at the concrete knowledge level is preserved, even if we map two concepts (sorts) s_1 and s_2 in $Sort_{concrete}$ into the same sort $\varphi(s_1) = \varphi(s_2)$ in the $Sort_{abstract}$. If a mapping φ satisfies the condition, then we can replace s_2 with s_1 in the proof of G whenever s_2 is used in it. So we need not distinguish s_1 and s_2 in the proof of G. In this sense we say that s_1 and s_2 are similar with respect to G. The sort mapping φ is a syntactic tool for representing the similarity w.r.t. the goal.

We thus focus on the ground of rule in seeking the appropriate similarities, and therefore disregard other knowledge not appearing in the proof of the goal.

The search space of our GDA algorithm is the set of all possible sort mappings. To reduce the computational cost in obtaining mappings that meets our condition, we postulate another requirement called a Similarity Inheritance Condition [8]:

> $\varphi(s_1) = \varphi(s)$ whenever $s_1 \leq s_2$ and $\varphi(s_2) = \varphi(s)$. (Suppose s_2 and s are similar and s_1 is a subclass of s_2. For s_1 inherits the properties of s_1 some of which are shared by s, s_1 would be also similar to s.)

The Similarity Inheritance Condition (SIC, for short), on the other hand, is deeply related to the structure of concept definitions. For the structure is generally considered as a terminological knowledge in the studies of Structural Inheritance Network, KL/ONE-like knowledge representations [11], and Description Logics, SIC can be regarded as a condition that comes from the terminological representation.

We have introduced SIC to find similarities more efficiently and to make our similarities conceptually clearer. So if we introduce some other condition originated from the terminological knowledge, then we would obtain more powerful effect in searching similarities and in making the detected similarities more persuasive. As such an additional condition, it would suffice to introduce the notion of roles between concepts, since it is a standard syntactic primitive in forming our terminological knowledge.

According to the standard studies of analogical reasoning [5; 15], analogy is a structural mapping between different domains that preserves structures of concepts in the domains. For the taxonomic hierarchy and the roles defines a structure of terminologies, we can say that the similarity constrained by the terminological structure is a structural analogy. Here we can state the new condition about roles.

> **Role Preservingness:** our similarity between concepts should preserve the value restrictions on roles in defining the concepts.

Combining this new requirement with SIC, we can say that

> Focusing on a part of our knowledge base from which our goals are derived, the similarity we have to find is supposed to preserve both the sub_class relationships (subsumption) and the value restrictions on roles. Consequently, those similar concept meeting our requirement will share the same terminological structure that is relevant to the goals.

Such a structural requirement for our similarities is strongly supported by the structure of terminology, the function of DL. DL is thus useful and powerful for not only a basic inference but also an extended inference such as analogy. As the previous study [4] has already pointed out, the syntax of DL is useful in inductive inference. This seems also true for analogical reasoning. This is the second reason why I am very interested in DL.

4 Knowledge Revision to handle exceptions

In this section, we discuss our third issue relating legal reasoning with DL.

Legal knowledge may alter, depending on case situations. We have to update the knowledge according to the change. In addition, it is well known that legal rules allow exceptions. In order to keep our legal knowledge base consistent, it seems necessary to handle exceptions in applying legal rules. A promising approach to the exception problem is to use knowledge revision operators, as presented in [16]. When DL is chosen as our language, a partial ordering known as subsumption relationship between concept descriptions [3] seems to be a key to find our solution. In the field of inductive learning, some algorithms for generating and revising inductive hypotheses have utilized such an ordering over first order terms in the case of FOL [2] and over concept graphs in the case of DL [4].

Inspired by these studies, we are now investigating a possibility to apply the subsumption ordering to the problem of exceptions, where the present framework assumes a subset of F-logic [10] in which the notion of concept graphs just coincides with the Ψ-term [1], provided we consider only attribute roles.

The major conclusion we have obtained in [7] is:

> Using the partial ordering (Ψ-term ordering) over concept graphs, we can define a notion of optimal revision for excluding exceptions. Moreover, given a set of positive and negative instances of the concept graphs. we have an algorithm to find the optimal one. The result is a set of revised rules consistent with the positive and negative instances. The original rule with exceptions are thus divided into several specialized rules to keep the optimality according to the minimal revision principle.

Thus it is an evidence for showing that the syntactic information represented by the ordering would be also useful for knowledge revision for exceptions. This is the third reason why I am learning DL.

Although we have not yet fully developed our approach for a wider class of descriptions, knowledge revision based on DL seems to produce various revision methodologies. This statement would be true as long as our legal knowledge base is terminological.

References

[1] Aït-Kaci,H. & Podelski,A. Towards a Meaning of LIFE, *Technical Report, PRL Presearch Report 11, DEC, Paris Research Lab.*, 1991

[2] Arimura,H. , Shinohara,T. & Otsuki,S. Polynomial Time Algorithm for Finding Finite Unions of Two Tree Pattern Languages, *Proceedings of the Second International Workshop on Nonmonotonic and Inductive Logic*, pp. 118-131. LNAI 659, Springer, 1993

[3] Borgida,A. & Patel-Schneider,P.F.: A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic, *Journal of Artificial Intelligence Research* 1, 277–308, 1994.

[4] Cohen,W.W.: The Learnability of Description Logics with Equality Constraints, *Machine Learning* 17, 169–199, 1994.

[5] Gentner,D.: Are Scientific Analogies Metaphors?, in *Metaphor: Problems and Perspectives*, The Harvester Press, 106-132, 1982.

[6] Haraguchi,M.: A Reasoning System for Legal Analogy *Machine Intelligence* Vol.14, 323–346, 1995.

[7] Ishikawa,Y & Haraguchi, M.: A Theory of Knowledge Revision using Multiple Generalization, *Proc. JSAI-SIGFAI workshop on inductive logic programming*, July 31–Aug.2, Sapporo, 1996.

[8] Kakuta,T. Haraguchi,M, Okubo,Y.: A Goal-Dependent Abstraction for Legal Reasoning by Analogy, submitted to Int'l J. of AI and Law, Feb. 1996. (The extended abstract in Japanese is found in Proc. 10th Annual Conf. of JSAI, 91–94, June. 1996)

[9] Kakuta,T. Haraguchi,M, Okubo,Y.: Legal Reasoning by Structural Analogy based on Goal- Dependent Abstractions, Ninth International Conference on Legal Knowledge-Based Systems, December 13, Tilburg University, the Netherlands, 1996.

[10] Kifer,M. , Lausen,G. & Wu,J. Logical Foundations of Object-Oriented and Frame-Based Languages, *Technical Report, Department of Computer Science SUNY at Stony Brook*, 1993

[11] Nebel,B.: Reasoning and Revision in Hybrid Representation Systems, Springer LNAI Vol. 422, 270 pages , 1990.

[12] Ohtake,Y. K.Nitta & et.al.: Legal reasoning system HELIC-II, Trans. Inf. Process. Soc. Jpn. Vol.35, No.6, pp. 986–996 (1994).

[13] Okubo,Y. & Haraguchi,M.: Constructing predicate mappings for goal-dependent abstraction *Proc. 5th Int'l Workshop on Algorithmic Learning Theory* Springer LNAI 872, 516–531, 1994.

[14] Tenenberg, J.D.: Abstracting first-order theories, in *Change of Representation and Inductive Bias*, D.P.Benjamin ed., Kluwer Academic Publishers, 67–79 1989.

[15] Winston,P.H.: Learning and Reasoning by Analogy, CACM, vol.2, no.3, 689–703, 1980.

[16] Wrobel,S.: Concept Formation and Knowledge Revision, *Kluwer Academic Publishers*, 1994

[17] Yamaguti,K.: Constructing a Legal Ontology from a General Ontology *Proc. Int'l Workshop on Application of Logic Programming to Legal Reasoning*, 73–80, 1994

[18] Yoshino,H, Haraguchi,M., Sakurai,S., Kagayama,S.: Towards a Legal Analogical Reasoning System: Knowledge Representation and Reasoning Methods, Proc. 4th ICAIL, 110–116, 1993.