Legal Assessment Using Conjunctive Queries

András Förhécz[1] and György Strausz[1]

Budapest University of Technology and Economics
Department of Measurement and Information Systems
Budapest, Hungary

Abstract. Using the Web Ontology Language (OWL) for knowledge representation in the legal domain is very promising but has some limitations. The language is complex thus hard to comprehend, still decidability results in a limited expressiveness which may introduce serious problems in modelling.
An aspect of limited expressiveness is the tree-model property of OWL, which can be overcome using rule formalisms or introducing variables, however losing decidability is not always acceptable.
We propose using conjunctive queries when modelling conditions of legal norms in the HARNESS architecture. Inference services required for modelling legislation and building legal assessment applications can be feasible using grounded queries resulting in a decidable formalism. Since most of the knowledge base remains within the limits of pure OWL2, we can benefit from consistency checking services of OWL2.

1 Introduction

Using the Web Ontology Language (OWL) for knowledge representation in the legal domain is very promising as different types of inference services are provided on top of a relatively expressive formalism, the description logic underlying OWL2.

The main benefits are strict semantics, consistency checking and feasible reasoning. In contrast to this, OWL is very complex thus hard to comprehend, resulting in a knowledge acquisition bottleneck. Remaining decidable also costs limited expressiveness which may introduce serious problems in modelling.

An aspect of limited expressiveness can be described with the tree-model property of OWL: only tree-like axioms are allowed (except for nominals, transitive properties and role inclusion axioms in OWL2). Complex structures cannot be described precisely due to the lack of cycles in axioms or predicates with arbitrary number of arguments: only unary (classes) and binary (properties) predicates are allowed.

Representing diamond-shaped structures is a frequently reoccuring problem. Suppose a sales contract with two actors – seller and customer – where the subject of the transaction should be joined to both the seller and the customer. Users familiar with rule formalisms tend to use rules or other extensions supporting variables to overcome these limitations, although, this way decidable satisfiability checking w.r.t. the T-Box is lost. In certain cases it is possible to represent cyclic structures using knowledge patterns and OWL2, as described in [1], nevertheless this solution cannot enforce `owl:sameAs` relations, only a custom property defined as a replacement.

An interesting approach for describing complex structures in OWL is the representation used in the HermiT reasoner [2]. Here the representation formalism is extended with description graphs for finite complex structures, where nodes and edges of graph-like structures are labelled with classes and properties respectively. Reasoning remains decidable but using arbitrary OWL axioms for the entities of complex structures is not allowed.

In this article we propose an alternative solution using conjunctive queries to solve legal assessment problems in the HARNESS[1] system. The next section introduces HARNESS, a legal knowledge-based system aimed at solving legal assessment problems. In the following two sections conjunctive queries are introduced and available inference services are described, including the case, when they are mixed with class expressions. In section 4.1. we show how to use conjunctive queries in HARNESS. The last section provides an overview of possible extensions and future plans.

2 Introducing HARNESS

A central task in legal knowledge-based systems is *legal assessment*: deciding whether some case is allowed or disallowed in a certain legal environment. In everyday situations a legal expert can help individuals to answer this sort of question, but due to the increasing size and complexity of legislation this process becomes more and more difficult, although transparency of jurisdictions would demand the opposite.

During the ESTRELLA[2] project an open platform was developed for legal knowledge technologies, including the Legal Knowledge Interchange Format (LKIF), a reference open source legal CMS called eXistrella, an argumentation engine Carneades and a DL-based inference system called HARNESS. The architecture of HARNESS enables solving different tasks including drafting or legal planning, although it is currently aimed solely at legal assessment.

HARNESS greatly exploits current Semantic Web technology by relying on formal ontologies and highly optimized DL reasoners. Legal assessment requires three distinct types of knowledge: a domain ontology, normative knowledge and case descriptions. The domain ontology defines the concepts and constraints in the field of interest and provides building blocks for defining individual cases. This ontology is a specialization of the LKIF Core ontology of basic legal concepts [3]. Normative knowledge describes regulations which govern the situations in question. Case descriptions underpin individual situations to be evaluated by HARNESS.

[1] Hybrid Architecture for Reasoning with Norms Exploiting Semantic web Services

[2] European project for Standardized Transparent Representations in order to Extend Legal Accessibility, IST-2004-027655, see `http://www.estrellaproject.org/`.

Each norm is expressed as a generic situation in which a state of action is qualified as undesirable, permitted or prescribed [4]. The situation itself is a conjunction of conditions, naturally expressed as a class expression in OWL, as specified in [5].

When modelling law in the HARNESS architecture using OWL the tree-model property of OWL hinders expressing complex situations. We will present an extension that will solve some of these issues: using *conjunctive queries* for specifying generic cases.

3 Using conjunctive queries

Conjunctive queries (CQ) are well known in database systems and have been standing in the focus of DL research for years now but not yet widely available in OWL applications. Practical results for complex DL languages have only appeared recently [6]. A possible syntax for such queries has just been defined in SPARQL-DL [7].

A conjunctive query is a conjunction of concept expressions of the form $C(t)$ and role expressions of the form $r(t, t')$ where C is a concept, r is a role and t, t' are terms, i.e. variables or individual names [6]. All variables are existentially qualified. These conditions are very similar to the body (condition) of SWRL rules. Introducing variables when specifying generic cases basically solves three different issues:

- We are no longer limited by the tree-model property of OWL, generic cases can express arbitrary relational structures.
- In a query, the values of variables can point at the case or part of the case the norm refers to. We can keep track of individuals when identifying obligations, permissions and violations.
- Using variables enlightens modelling. Most knowledge experts are familiar with variables and it is easier for them to specify the condition with conjunctive queries. When additional expressiveness is not required, the CQ can be automatically transformed into an OWL class expression [8].

The following example demonstrates the usage of CQs. In Section 16. paragraph (2) in the Hungarian Law on Duties[3] an exemption is specified for paying duties on a land received as a gift:

> "In order to verify completion of the construction of the residential house [...] the state tax authority shall contact the competent building authority [...] If the building authority provides a certificate in proof of the occupancy permit issued to the name of the property owner, the state tax authority shall cancel the duty assessed, but suspended in respect of payment."

[3] Hungarian Law on Duties, Act XCIII of 1990, only available in Hungarian, `http://net.jogtar.hu/jr/gen/hjegy_doc.cgi?docid=99000093.TV#pr123`

The condition of the exemption can be formalized using conjunctive queries the following way. If a gift ($?g$) is a *plot of land*, and a building ($?b$) has been built on it, which is a *residential house*, and an *occupancy permit* ($?p$) was issued to the name of the *donee* ($?d$), the generic case is fulfilled:

$$\begin{aligned} GC_{S16\_2} \equiv\ & Donation(?t) \wedge donee(?t, ?d) \\ & \wedge subject(?t, ?g) \wedge PlotOfLand(?g) \\ & \wedge built\_onto(?b, ?g) \wedge ResidentialHouse(?b) \\ & \wedge permit\_issued(?b, ?p) \wedge OccupancyPermit(?p) \\ & \wedge issued\_to(?p, ?d) \end{aligned} \tag{1}$$

A conjunctive query can be represented by a graph where each variable in the query give rise to a node in the graph. Concept names appear as node labels, role names as edges at the appropriate variables in the graph. Figure 1 represents the graph for the CQ shown above.

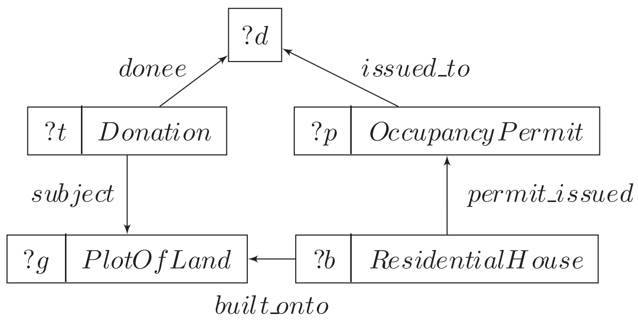

Fig. 1. Graph representation of a conjunctive query

4 Inferences and results

Just like OWL classes, different inference services can be implemented to conjunctive queries:

query entailment is the decision problem to answer whether a query is true in all models of a knowledge base.

query answering is the problem of finding all answer tuples for a query. If the entailment is false, there will be no answers. Otherwise, there may be one or more tuples fulfilling the constraints described in a query.

satisfiability is to decide if a knowledge base has at least one model in which the query is true. When building a model and respective queries, a non-satisfiable query may indicate inconsistency in the model, as the corresponding legal condition will never be fulfilled.

subsumption of CQs can be defined in a similar manner as class subsumption: with respect to a knowledge base $\mathcal{K}$ a query Q_1 subsumes the query Q_2 if in all models where Q_2 is true, Q_1 is also true. A more specific condition may correspond to a norm with higher priority based on *lex specialis*.

Not all of these problems are solved for the description logic underlying OWL2. Satisfiability can be easily answered using a DL reasoner, but it is still an open issue whether the other problems are decidable for the DL $\mathcal{SHOIN}(\mathcal{D})$. Latest results showed that query entailment is decidable for $\mathcal{SHIQ}$ [9] and $\mathcal{SHOQ}$ [10] which are slightly restricted sublanguages of OWL2.

However in the general interpretation variables in a CQ are not required to correspond to a named individual in the ABox. For so-called *non-distinguished variables* only the existence of a suitable element is required in the model, and *answer variables* are required to have a corresponding named individual. This is important as in the restricted closed-world interpretation of CQs we only use answer variables, and then all inference problems are decidable in OWL2.

A subsumption hierarchy of CQs can be derived the same way as for OWL classes. Conjunctive queries are a generalization of OWL class expressions, as all class expression can be trivially transformed to an atomic CQ with one variable:

$$C \rightarrow Q(x) \equiv C(x)$$

As a result subsumption can be defined across CQs and OWL named classes, and hierarchy of CQs and classes can be merged. As an example for the CQ in equation 1: $GC_{S16\_2} \sqsubseteq Donation$.

Satisfiability of conjunctive queries can be derived from subsumption the same way as for OWL classes, by defining the always unsatisfiable CQ:

$$Q_{\perp}(x) \equiv \perp(x)$$

$$Q(\ldots) \text{ is unsatisfiable} \Leftrightarrow Q(\ldots) \sqsubseteq Q_{\perp}(x)$$

Subsumption relations across CQs and OWL classes are important for designing tool support and knowledge engineering methodology. Conjunctive queries are not yet natively supported by major ontology editors but can be integrated into e.g. class hierarchy in a relatively straightforward manner giving confidence to knowledge engineers.

4.1 Application in HARNESS

In legal inference we have two distinct modelling issues: creating a domain ontology for enforcing valid case descriptions and formalizing normative knowledge in legal assessment. In the assessment part we are using the HARNESS architecture [5] providing definitions for generic cases (GC). A generic case represents the situation describing the condition part of a norm.

The domain ontology must be consistent and all kind of inferences (including full consistency) are required, OWL is an adequate formalism. With GC descriptions, however, the only inferences required are hierarchy of GCs (is one description more general than another?) and case entailment (does a case fulfill all conditions of a GC?). The problem of modelling complex structures generally occur with GCs, so we will propose to use conjunctive queries in specifying generic cases.

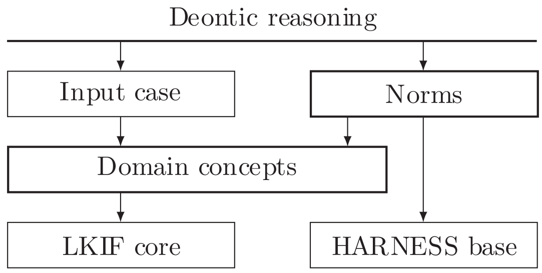

Fig. 2. Interactions of ontology modules in HARNESS

We can use conjunctive queries to describe generic cases while the rest of the model is still specified in OWL2. In Figure 2 parts of a general HARNESS knowledge base are shown. *LKIF-core* and *HARNESS base* are standard ontology modules: providing basic legal concepts, the base class for norms and deontic operators. *Domain concepts* include terminological knowledge for specifying input case descriptions. The normative knowledge is provided in part *Norms*, allowing conjunctive queries for modelling conditions. Consistency of an input case specification can be verified using conventional DL tools and only evaluating the deontic reasoner requires conjunctive query evaluation.

Conjunctive queries are appropriate because inference services are available to cover the tasks required in HARNESS:

- *query answering* matches input case descriptions with relevant norms,
- *subsumption relations* provide exceptions (lex specialis) for norms in the model and
- *satisfiability* ensures that each norm is consistent with the domain model.

An experimental implementation for the closed-world interpretation (using only answer variables) has been provided. The reasoner can use any DL reasoner (black-box reasoning) or Pellet[4] and its highly optimized algorithms. The CQ reasoner can be accessed from Protégé 4[5] as a plug-in and can be selected instead other DL reasoners.

When using HARNESS, legal knowledge bases with conjunctive queries are supported. The generic situations in the normative part of the knowledge base

[4] Pellet: an open source OWL 2 DL reasoner developed by Clark & Parsia, LLC `http://clarkparsia.com/pellet/`

[5] Protégé 4 ontology editor, `http://protege.stanford.edu/`

can be expressed both using OWL class expressions or conjunctive queries. Generic situations can be reviewed in a combined hierarchy following *lex specialis* relations and also showing all cases satisfying the condition.

5 Future plans

An important feature of legal expert systems is the ability to provide reliable and comprehensible explanations for inference results. For OWL this can be achieved using a recent feature of the Pellet reasoner: laconic justifications [11, 12]. These are minimal set of axioms supporting a single conclusion, extracting the piece of information required for understanding a single issue. As conjunctive queries are handled by an additional reasoning mechanism, explanation services should be extended to support justifications in these legal knowledge bases.

Unfortunately description logics are hard to comprehend for the casual users, so when non-expert users should be able to interpret explanations, OWL axioms have to be translated to natural language or a graphical representation. The former can be achieved with NLP tools like ROO Rabbit [13] or Ace View [14]. We already took steps on adopting these services to handle queries and translate them to our target language, Hungarian.

References

1. Hoekstra, R., Breuker, J.: Polishing diamonds in OWL 2. In: EKAW. (2008) 64–73
2. Motik, B., Grau, B.C., Sattler, U.: Structured objects in OWL: Representation and reasoning. In: WWW. (2008) 555–564
3. Hoekstra, R., Breuker, J., Bello, M.D., Boer, A.: The LKIF core ontology of basic legal concepts. In: LOAIT. (2007) 43–63
4. Valente, A.: Legal knowledge engineering: A modelling approach. PhD thesis, University of Amsterdam (1995)
5. van de Ven, S., Hoekstra, R., Breuker, J., Wortel, L., El-Ali, A.: Judging amy: Automated legal assessment using OWL 2. In: OWLED. (2008)
6. Glimm, B.: Querying Description Logic Knowledge Bases. PhD thesis, The University of Manchester, Manchester, United Kingdom (2007)
7. Sirin, E., Parsia, B.: SPARQL-DL: SPARQL query for OWL-DL. In: 3rd OWL Experiences and Directions Workshop (OWLED-2007). (2007)
8. Gasse, F., Haarslev, V.: DLRule: A rule editor plug-in for Protege. In: OWL: Experiences and Directions (OWLED). (2008)
9. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive query answering in the description logic $\mathcal{SHIQ}$. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007). (2007)
10. Glimm, B., Horrocks, I., Sattler, U.: Conjunctive query entailment for shoq. In: Description Logics. (2007)
11. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in OWL. In: International Semantic Web Conference. (2008) 323–338
12. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: ISWC/ASWC. (2007) 267–280

13. Dimitrova, V., Denaux, R., Hart, G., Dolbear, C., Holt, I., Cohn, A.G.: Involving domain experts in authoring OWL ontologies. In Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T.W., Thirunarayan, K., eds.: International Semantic Web Conference. Volume 5318 of Lecture Notes in Computer Science., Springer (2008) 1–16
14. Kaljurand, K.: ACE View — an ontology and rule editor based on Attempto Controlled English. In: OWLED. (2008)