Practical Fixed-Domain Reasoning for Description Logics – Extended Abstract

Sarah Alice Gaggl, Sebastian Rudolph, and Lukas Schweizer

Technische Universität Dresden, Computational Logic Group
firstname.lastname@tu-dresden.de

In this extended abstract, we report on our work on fixed-domain reasoning published at the 22nd European Conference on Artificial Intelligence [6].

Introduction. The Web Ontology Language OWL [13] comes with comprehensive modeling tool support. This sometimes leads to situations where OWL is the modeling paradigm chosen over other formalisms, even if the application scenario does not match the typical usage of this language. For example, problems of a constraint-satisfaction nature do not go well with OWL's standard semantics which allows for models of arbitrary size.

Example 1 *Assume we want to find out if, in a given graph (V, E), a Hamiltonian path from vertex v_α to vertex v_ω exists. Using one individual name v for every vertex $v \in V$, one might come up with the axioms (in description logic notation)* $\texttt{noedge}(v, v')$ *whenever* $(v, v') \notin E$, *as well as* $\top \sqsubseteq \leq 1\texttt{edgeonpath}.\top \sqcap \leq 1.\texttt{edgeonpath}^-.\top$ *and* $\neg\{v_\alpha\} \sqsubseteq \exists\texttt{edgeonpath}^-.\top$ *and* $\neg\{v_\omega\} \sqsubseteq \exists\texttt{edgeonpath}.\top$, *requiring that all nodes have exactly one incoming and one outgoing edge that is on the Hamiltonian path, except for v_α which just has an outgoing and v_ω which just has an incoming such edge. The axiom* $\{v_\alpha\} \sqsubseteq (\exists\texttt{edgeonpath}.)^{|V|-1}\{v_\omega\}$ *expresses that there is an* edgeonpath*-path of length $|V| - 1$ from v_α to v_ω. The axiom* $\mathsf{Dis}(\texttt{edgeonpath}, \texttt{noedge})$ *ensures that the path can never go along "no-edges". Under the usual semantics, however, this set of axioms is always satisfiable since models may contain "anonymous" elements that do not correspond to any of the node individual names. The given formalisation is only appropriate if we restrict to models with domain V.*

To overcome these shortcomings, we propose *fixed-domain reasoning* for description logics (DLs), imposing a non-standard model-theoretic semantics that restricts the domain to an explicitly given, fixed finite set.

Models over Fixed Domains. In DLs, models can be of arbitrary cardinality. In many applications, however, the domain of interest is known to be finite. In fact, restricting reasoning to models of finite domain size (called *finite model reasoning*, a natural assumption in database theory), has already become the focus of intense studies in DLs [4, 11]. As opposed to assuming the domain to be merely finite (but of arbitrary, unknown size), we consider the case where the domain has an *a priori known cardinality* and use the term *fixed domain*.

Definition 1 (Fixed-Domain Semantics). *Given a non-empty finite set Δ of individual names, called* fixed domain*, an interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ is said to*

be Δ-fixed *(or just* fixed*, if* Δ *is clear from the context), if* $\Delta^{\mathcal{I}} = \Delta$ *and* $a^{\mathcal{I}} = a$ *for all* $a \in \Delta$*. Accordingly, for a DL knowledge base* $\mathcal{K}$*, we call an interpretation* $\mathcal{I}$ *a* Δ-model *of* $\mathcal{K}$*, if* $\mathcal{I}$ *is a* Δ*-fixed interpretation and* $\mathcal{I} \models \mathcal{K}$*. A knowledge base* $\mathcal{K}$ *is called* Δ-satisfiable *if it has a* Δ*-model.*

Enumeration of Δ***-Models.*** As mentioned before, our approach aims at scenarios like constraint satisfaction problems where a knowledge base is a formal problem description for which each model represents one solution; in particular the domain is part of the problem description and hence fixed a-priori. Then, retrieval of one, several, or all models is a natural task, as opposed to merely checking model existence. We let *model enumeration* denote the task of making the Δ-models of a knowledge base explicit.

Axiomatization of Δ***-Models.*** When introducing a new semantics for some logic, it is worthwhile to ask if existing reasoners can be used. Obviously, assuming $\Delta = \{a_1, \ldots, a_n\}$, adding the DL axiom $\top \sqsubseteq \{a_1, \ldots, a_n\}$ as well as the set of inequality axioms containing $a_i \neq a_j$ with $i < j$ to $\mathcal{K}$ will (up to isomorphism) rule out all models of $\mathcal{K}$, not having Δ as their domain. Denoting these additional axioms with $\mathcal{FD}_\Delta$, we find that $\mathcal{K}$ is Δ-satisfiable iff $\mathcal{K} \cup \mathcal{FD}_\Delta$ is satisfiable under the classical DL semantics. Consequently, any off-the-shelf $\mathcal{SROIQ}$ reasoner can be used for fixed-domain reasoning, at least when it comes to the classical reasoning tasks.

Complexity Analysis. We report on complexities of classical reasoning tasks under the fixed-domain semantics. Note that, next to the size of the considered knowledge base the size of the domain $|\Delta|$ contributes to the input size of the reasoning problems considered.

The combined complexity of standard reasoning in $\mathcal{SROIQ}$ is known to be N2ExpTime-complete, both for arbitrary models and finite models [9]. Restricting to fixed domains leads to a drastic drop in complexity. Contrarily, imposing fixed domains on (allegedly) inexpressive fragments such as $\mathsf{DL\text{-}Lite}_{\mathsf{core}}$, turns reasoning into a harder problem.

Let $\mathsf{DL}_{\mathsf{min}}$ be a minimalistic description logic that merely allows TBox axioms of the form $A \sqsubseteq \neg B$, with $A, B \in \mathsf{N}_C$. Moreover, only atomic assertions of the form $A(a)$ and $r(a, b)$ are admitted.

Theorem 1. *Fixed-domain satisfiability checking in any language between* $\mathsf{DL}_{\mathsf{min}}$ *and* $\mathcal{SROIQ}$ *is* NP*-complete.*

We next consider the complexity of query entailment for DLs. Again, we will notice a very uniform behavior over a wide range of DLs and query types.

Bounded-arity Datalog queries over DLs are rather expressive, they subsume many of the prominent query classes in knowledge representation and databases, including (unions of) conjunctive queries, positive queries, (unions of) conjunctive 2-way regular path queries [5], positive 2-way regular path queries, (unions of) conjunctive nested 2-way regular path queries [1], and regular queries as defined in [10]. We obtain the following theorem.

Theorem 2. *For any class of queries subsuming conjunctive queries and subsumed by bounded-arity Datalog queries and any DL subsuming* $\mathsf{DL}_{\mathsf{min}}$ *and subsumed by* $\mathcal{SROIQ}$*, the combined complexity of fixed-domain query entailment is* Π_2^{P}*-complete.*

Practical Fixed-Domain Reasoning. When axiomatizing the fixed-domain semantics, available OWL reasoners struggle with standard reasoning, and we supported this statement with an evaluation. Thus, a more viable approach is required for realizing fixed-domain reasoning. To this end, we propose an encoding of arbitrary $\mathcal{SROIQ}$ knowledge bases into *answer set programming* (ASP) [2]. This allows us to use existing ASP machinery to perform both standard reasoning as well as the non-standard tasks *model enumeration* and query entailment quite elegantly.

We now briefly sketch how reasoning tasks w.r.t. the fixed-domain semantics can be encoded by ASP. Intuitively, the set of all Δ-interpretations defines a search space, which can be traversed searching for Δ-models, guided by appropriate constraints. We thus propose a translation $\Pi(\mathcal{K}, \Delta)$ for any $\mathcal{SROIQ}$ knowledge base $\mathcal{K}$; i.e. $\Pi(\mathcal{K}, \Delta) = \Pi_{\mathrm{gen}}(\Delta) \cup \Pi_{\mathrm{chk}}(\mathcal{K})$, consisting of a generating part $\Pi_{\mathrm{gen}}(\Delta)$ that defines all potential candidate interpretations using so-called guessing rules, and a constraining part $\Pi_{\mathrm{chk}}(\mathcal{K})$ containing only integrity constraints (i.e., rules with empty heads) that rules out interpretations violating axioms in $\mathcal{K}$.

We implemented our translation based approach as an open-source tool – named `Wolpertinger`.[1] The obtained logic programs can be evaluated with most modern ASP solvers. However, the evaluation was conducted using `Clingo` [7] for grounding and solving, since it is currently the most prominent solver leading the latest competitions [3].

For satisfiability checking, we conducted some preliminary tests, comparing our tool to the popular `HermiT` and `Konclude` reasoners [8, 12]. Both are full OWL 2 DL reasoners and are leading the latest competitions. We are aware that these reasoners are designed for and optimized toward open-world scenarios, the conducted tests shall merely show the feasibility of our approach in comparison to standard DL reasoners using the previously discussed axiomatization.

Among other experiments, we created a knowledge base modeling fully and correctly filled Sudokus, beginning with a 6×6 field, consisting of 64 individuals, 13 concept names and 1 role name, then extending the size to a 9×9 field featuring 108 named individuals. While `HermiT` & `Konclude` were still able to detect satisfiability for the 6×6 case, invoking a satisfiability test on the 9×9 case did not yield any answer within 30 minutes. In both cases `Wolpertinger` detected satisfiability with reasonable runtimes of below 10 seconds.

For model enumeration, we used the knowledge base for the 9×9 Sudoku and turned the task into generating new Sudoku instances. We observed that, besides

[1] `https://github.com/wolpertinger-reasoner/Wolpertinger`

a constant time of around 6 seconds required for grounding, even requesting 10^6 models is reasonably efficient, i.e., it requires less than 5 minutes.

Conclusion and Future Work. The fixed-domain semantics allows to confine modelhood of interpretations to models of the right form, for OWL ontologies which represent constraint-type problems. Although OWL still imposes some restrictions regarding expressivity (e.g., by restricting the arity of the used predicates to 1 and 2), we argue that quite large and involved problem scenarios can be modeled by OWL ontologies. Clearly, more comprehensive evaluations of our system with respect to such ontologies remain as imperative issue. Moreover, translations of fixed-domain reasoning problems into other formalisms are conceivable, including pure CSP languages or even SAT, which would have to be implemented and compared against the ASP approach.

Another interesting strand of research would be to consider extensions of the source formalism, e.g. by non-monotonic features. As ASP itself is a non-monotonic logic programming formalism, rule-based extensions of OWL – monotonic or non-monotonic – should be straightforward to accommodate.

Finally, we plan to incorporate typical ontology engineering tasks such as explanation and axiom pinpointing into our ASP-based framework.

References

1. M. Bienvenu, D. Calvanese, M. Ortiz, and M. Simkus, 'Nested regular path queries in description logics', in *Proc. KR (2014)*.
2. Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński, 'Answer set programming at a glance', *Commun. ACM*, **54**(12), 92–103, (2011).
3. F. Calimeri, M. Gebser, M. Maratea, and F. Ricca, 'Design and results of the 5th answer set programming competition', *Artif. Intell.*, **231**, 151–181, (2016).
4. D. Calvanese, 'Finite model reasoning in description logics', in *Proc. DL (1996)*.
5. D. Calvanese, T. Eiter, and M. Ortiz, 'Regular path queries in expressive description logics with nominals', in *Proc. IJCAI (2009)*.
6. S. A. Gaggl, S. Rudolph, and L. Schweizer, 'Fixed-domain reasoning for description logics', in *Proc. ECAI (2016)*.
7. M. Gebser, B. Kaufmann, R. Kaminski, M. Ostrowski, T. Schaub, and M. T. Schneider, 'Potassco: The Potsdam Answer Set Solving Collection', *AI Communications*, **24**(2), 107–124, (2011).
8. B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, 'HermiT: an OWL 2 reasoner', *Journal of Automated Reasoning*, **53**(3), 245–269, (2014).
9. Y. Kazakov, '$\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$', in *Proc. KR (2008)*.
10. J. L. Reutter, M. Romero, and M. Y. Vardi, 'Regular Queries on Graph Databases', in *Proc. ICDT (2015)*.
11. S. Rudolph, 'Undecidability results for database-inspired reasoning problems in very expressive description logics', in *Proc. KR (2016)*.
12. A. Steigmiller, T. Liebig, and B. Glimm, 'Konclude: System description', *Journal of Web Semantics*, **27**, 78–85, (2014).
13. W3C OWL Working Group, *OWL 2 Web Ontology Language: Document Overview*, W3C Recommendation, 2009.