

# From Horn- $SRIQ$ to Datalog: A Data-Independent Transformation that Preserves Assertion Entailment\*

David Carral<sup>1</sup>, Larry González<sup>1</sup>, and Patrick Koopmann<sup>1,2</sup>

<sup>1</sup> Knowledge-Based Systems Group, TU Dresden, Dresden, Germany

<sup>2</sup> Institute for Theoretical Computer Science TU Dresden, Germany

Assertion retrieval (AR)—i.e., the task of inferring all implicit assertions from a Description Logics (DL) knowledge base (KB)—is an important reasoning task with many applications in knowledge representation and data management. To access data efficiently, one approach is to rewrite the ontology into Datalog, and then use powerful Datalog engines to compute implicit entailments. Existing rewriting techniques support DL languages from  $\mathcal{ELH}$  to Horn- $SHIQ$  [3]. We present the first such transformation for Horn- $SRIQ_{\sqcap}$ , an expressive DL that allows for the use of the role chain constructor. Furthermore, we show that the resulting method for assertion retrieval is worst-case optimal for  $\mathcal{ELH}$ , Horn- $SHIQ$ , and Horn- $SRIQ_{\sqcap}$ . We empirically show that the use of Datalog rewritings can outperform state-of-the-art reasoners and that we can construct rewritings of moderate sizes for many real-world ontologies.

To give a formal intuition about our approach, our method computes an *AR-rewriting* in the following sense.

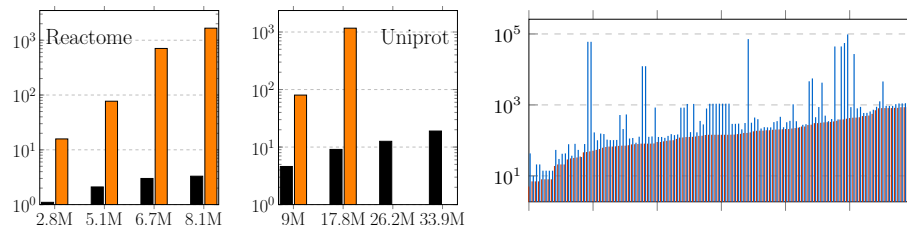
**Definition 1.** Let  $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$  be a Horn- $SRIQ_{\sqcap}$  KB. A Datalog rule set  $\mathcal{R}_{\mathcal{T}}$  is an AR-rewriting for  $\mathcal{T}$  iff, for every ABox  $\mathcal{A}$  and assertion  $\alpha$  defined over the signature of  $\mathcal{T}$ ,  $\langle \mathcal{T}, \mathcal{A} \rangle$  and  $\langle \mathcal{R}_{\mathcal{T}}, \mathcal{A} \rangle$  are equi-satisfiable and  $\langle \mathcal{T}, \mathcal{A} \rangle \models \alpha$  iff  $\mathcal{K}_{\mathcal{O}} = \langle \mathcal{R}_{\mathcal{T}}, \mathcal{A} \rangle \models \alpha$ .

A well-known way of characterising entailments from KBs is the chase [2]. An alternative characterisation of Definition 1 is that  $\mathcal{R}_{\mathcal{T}}$  is an AR-rewriting of  $\mathcal{O}$  if the chase of  $\mathcal{K}_{\mathcal{O}}$  (denoted with  $\mathcal{K}_{\mathcal{O}}^{\infty}$ ) coincides with the chase of  $\mathcal{O}$  (denoted with  $\mathcal{O}^{\infty}$ ) on all assertions over the signature of  $\mathcal{T}$ . The challenge in constructing Datalog AR-rewritings is that facts in  $\mathcal{O}^{\infty}$  may contain an unbounded amount of *nulls* (i.e., “unnamed” terms that are introduced to satisfy existential restrictions), whereas Datalog rules can only derive facts containing a fixed amount of constants. To replicate assertion entailments in  $\mathcal{K}_{\mathcal{O}}^{\infty}$ , we encode the relevant information about the null successors of an individual in  $\mathcal{O}^{\infty}$  by introducing fresh concepts and roles. Here, we benefit from a well-known automata technique [4,5] to trace the paths of complex roles in the chase.

To ascertain when information about one of these fresh predicates needs to be used in  $\mathcal{K}_{\mathcal{O}}$ , we make use of the calculus from [3], which is also used to infer

---

\* The full paper will be published in the proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI’19) [1].



**Fig. 1.** On the left: time in seconds for RDFox (dark) and Konclude (bright), each over four ABoxes with increasing numbers of assertions. On the right, Size of TBoxes and their rewritings.

axioms relevant to our Datalog program. Since this calculus was originally designed for Horn-*SHIQ*, we first need to extend our input TBox  $\mathcal{T}$  to a TBox  $\mathcal{T}_+$  in which the behaviour of complex role inclusion axioms is sufficiently simulated.

To show practical feasibility of our approach, we implemented a prototype and performed two different experiments. We first compared the performance of solving AR using our Datalog rewritings versus using the DL reasoner *Konclude* [8]. We considered two real-world, data-intensive ontologies from the biological domain, *Reactome* and *Uniprot* [9] which we normalised and enriched with 3 and 5 complex role inclusion axioms, respectively. The resulting TBoxes contain 485 and 304 axioms. The rewritten Datalog programs contain 539 and 367 rules, and were computed in 221 and 182 seconds. For each ontology, we considered ABoxes of various sizes, generated by sampling the real-world ABoxes using the method by Zhou et al. [9]. We use RDFox [6] as Datalog engine for computing the chase of our rewritings, and compare its performance with that of Konclude. Figure 1 shows the wall-clock times measured in this experiment, ignoring the time used for parsing and loading, in logarithmic scale.

In our second experiment, we computed rewritings for 121 selected TBoxes from MOWLCorp [7]. Figure 1 shows the number of axioms of the input ontologies (blue bars) and the number of rules of the Datalog rewritings (red bars). For some ontologies, the rewritings got substantially larger. This was expected, and in theory unavoidable, due to the double exponential time complexity of assertion entailment in Horn-*SRIQ*<sub>Γ</sub>: for Datalog, this complexity is only polynomial (since predicate arity is bounded), which is why our rewritings are in the worst case double exponential in the size of the input. Our evaluation confirms that these blow-ups are not only of theoretical nature, but may actually happen for some real-world ontologies. However, in most cases, the size of the computed Datalog rule sets was still of comparable dimensions: in 59% of cases, the increase was at most by 100%, and in 74% of cases, it was at most by 200%.

*Acknowledgements.* The first two authors are funded by Deutsche Forschungsgemeinschaft in project number 389792660 (TRR 248, Center for Perspicuous Systems) and Emmy Noether grant KR 4381/1-1 (DIAMOND). The third author is funded by DFG in the CRC 912 (HAEC).

## References

1. Carral, D., González, L., Koopmann, P.: From Horn-*SRIQ* to datalog: A data-independent transformation that preserves assertion entailment. In: Proceedings of the 33rd Conference on Artificial Intelligence (AAAI 2019) (January 2019)
2. Deutsch, A., Nash, A., Rommel, J.B.: The chase revisited. In: Proc. PODS' 08. ACM SIGMOD-SIGACT-SIGART. ACM (2008)
3. Eiter, T., Ortiz, M., Šimkus, M., Tran, T., Xiao, G.: Query rewriting for Horn-*SHIQ* plus rules. In: Proc. AAAI'12. AAAI Press (2012)
4. Horrocks, I., Kutz, O., Sattler, U.: The irresistible *SRIQ*. In: Proc. OWLED'05. CEUR Workshop Proceedings, vol. 188. CEUR-WS.org (2006)
5. Kazakov, Y.: An extension of complex role inclusion axioms in the description logic *SROIQ*. In: Proc. IJCAR'10. Lecture Notes in Computer Science, vol. 6173, pp. 472–486. Springer (2010)
6. Motik, B., Nenov, Y., Piro, R., Horrocks, I., Olteanu, D.: Parallel materialisation of Datalog programs in centralised, main-memory RDF systems. In: Proc. AAAI'14. pp. 129–137. AAAI Press (2014)
7. Parsia, B., Matentzoglou, N., Gonçalves, R.S., Glimm, B., Steigmiller, A.: The OWL reasoner evaluation (ORE) 2015 competition report. *J. of Autom. Reasoning* **59**(4), 455–482 (2017)
8. Steigmiller, A., Liebig, T., Glimm, B.: Konclude: System description. *J. of Web Semantics* **27**, 78–85 (2014)
9. Zhou, Y., Cuenca Grau, B., Nenov, Y., Kaminski, M., Horrocks, I.: PAGOdA: Pay-as-you-go ontology query answering using a Datalog reasoner. *J. of Artif. Intell. Res.* **54**, 309–367 (2015)