Reconciling SHACL and Ontologies: Semantics and Validation via Rewriting (Extended Abstract)

Shqiponja Ahmetaj[1], Magdalena Ortiz[2], Anouk M. Oudshoorn[1] and Mantas Šimkus[2]

[1]Technical University of Vienna, Karlsplatz 13A, 1040 Wien, Austria
[2]Umeå University, Universitetstorget 4, 901 87 Umeå, Sweden

Abstract
This extended abstract summarizes our recent work [1] on SHACL validation in the presence of OWL 2 QL ontologies. To overcome the challenge posed by the non-monotonic behavior of SHACL constraints, we propose a new intuitive validation semantics and a rewriting algorithm that embeds the effects of the ontological axioms into the SHACL constraints. We analyze the complexity of validation in this setting.

Keywords
SHACL, OWL 2 QL, validation, rewriting, complexity

1. Introduction

SHACL and OWL are two prominent W3C standards for managing RDF data, the graph-based data model of the Web. They were specifically designed to target two different issues. OWL was standardized in parallel with RDF to address information incompleteness of RDF data by means of ontological axioms that complete the data with missing information. OWL and its profiles are based on *Description Logics (DLs)* [2] and make the open-world assumption (OWA), which intuitively means that the data only presents a partial description of the domain of interest and missing facts may also be true. RDF was soon adopted by increasingly many applications and making decisions based on correct data became particularly crucial.

To check the correctness of RDF data, W3C proposed the so-called *Shapes Constraint Language (or SHACL)* [3], a machine-readable constraint language for describing and validating RDF graphs. Unlike OWL, SHACL operates under the closed-world assumption (CWA) and assumes completeness of data. SHACL specifies the notion of a *shapes graph*, which consists of a set of shape constraints paired with the so-called *targets*, which is a selection of nodes of the data graph that must be validated against the constraints. The precise semantics of SHACL in the presence of recursion was not described in the W3C standard, which led to recent works that propose semantics based on first-order logic and logic programming [4, 5, 6].

Combining SHACL and OWL into a setting that allows to perform RDF data validation while taking into account the implicit facts inferred using an OWL ontology is a relevant

DL 2023: 36th International Workshop on Description Logics, September 2–4, 2023, Rhodes, Greece
shqiponja.ahmetaj@tuwien.ac.at (S. Ahmetaj); magdalena.ortiz@umu.se (M. Ortiz); anouk.oudshoorn@tuwien.ac.at (A. M. Oudshoorn); mantas.simkus@umu.se (M. Šimkus)
0009-0006-4638-5948 (S. Ahmetaj); 0000-0002-2344-9658 (M. Ortiz); 0009-0006-4638-5948 (A. M. Oudshoorn); 0000-0003-0632-0294 (M. Šimkus)

but challenging problem. Indeed, the W3C SHACL specification envisions graph validation in the presence of OWL entailment but does not provide guidance on how to realize this. To our knowledge, this has only been addressed in [7], which considers *positive* SHACL constraints only. To see the benefits of taking into account ontologies when performing validation, consider an example of a simplistic database of pet owners containing the facts $hasWingedPet(linda, blu)$, $Bird(blu)$, $PetOwner(john)$, $hasPet(john, ace)$ and consider the simple constraint $petOwnerShape \leftarrow PetOwner \vee \exists hasPet$ with the target $petOwnerShape(linda)$, which asks to verify whether $linda$ is a pet owner. Clearly, one would expect the input data to validate $linda$ as a pet owner given that she has a winged pet. However, this is not the case since the setting is missing the background knowledge that owning a winged pet implies owning a pet. The latter can be expressed through the ontological axiom $hasWingedPet \sqsubseteq hasPet$, which would allow us to obtain the desired validation result.

Identification of a proper semantics in this setting requires integrating the OWA of OWL and the CWA of SHACL. There are several proposals by the DL and database communities to relax the OWA and combine it with CWA [8, 9, 10, 11]. Another challenge when defining a validation semantics is dealing with the non-monotonic behavior of SHACL constraints due to the presence of negation. Roughly speaking, adding facts to the input data graph may cause a previously valid setting to become invalid. Such non-monotonic behavior is known when combining ontologies and negation in the so-called conjunctive queries or database constraints (see e.g., [12, 9, 13]). The main contributions of our work in [1] can be summarized as follows:

◦ We present a novel notion of SHACL validation in the presence of a DL-Lite$_\mathcal{R}$ ontology, the logic underlying OWL 2 QL [14]. Specifically, we consider *stratified* SHACL constraints, which support a limited form of recursion (limiting the interaction between recursion and negation). Our notion of stratification is derived from the well-known class of stratified logic programs [15]. For instance, the constraint $petOwnerShape \leftarrow PetOwner \vee \exists hasPet.\neg petOwnerShape$ is not stratified since the shape name $petOwnerShape$ depends *negatively* on itself. On the other hand, the constraints $petOwnerShape \leftarrow PetOwner \vee \exists hasPet.petShape$ and $petShape \leftarrow Pet \wedge \neg WildAnimal$ are stratified. We note that the current SHACL standard defines the semantics only for non-recursive constraints, leaving the recursive case open.

◦ Since SHACL constraints involve negation, defining a semantics of validation in the presence of ontologies is challenging. In our approach, knowledge stemming from the ontology is included by completing the input data graph with additional facts to satisfy the ontological axioms. We adopt a completion that is *austere* in the sense that only a minimal amount of new facts is added at each step of the procedure. Validation of constraints over a data graph in the presence of an ontology is defined as validation of the constraints in the possibly infinite *austere canonical model* that we introduce. We discuss this in more detail in Section 2.

◦ Since validation in this paper is defined over the (potentially infinite) austere canonical model, its computational complexity is not obvious. We prove that this problem is decidable and is PTime-complete in data complexity. This coincides with the complexity of stratified constraints over plain data graphs [6], and shows that adding a DL-Lite$_\mathcal{R}$ ontology actually does not incur additional costs in data complexity. This is different for combined complexity, which turns out to be ExpTime-complete. The high combined complexity is somewhat surprising, since individually standard reasoning in DL-Lite$_\mathcal{R}$ ontologies and validation of stratified SHACL constraints over plain data graphs are tractable in combined complexity.

∘ Our upper bounds on complexity follow from a *constraint rewriting technique* that we introduce in this paper. We design an inference procedure that takes as input an ontology $\mathcal{T}$ together with a set $\mathcal{C}$ of stratified constraints, and produces as output a new set $\mathcal{C}_\mathcal{T}$ of stratified constraints such that $\mathcal{C}_\mathcal{T}$ alone is equivalent to the pair of $(\mathcal{T}, \mathcal{C})$, i.e. for validation, $\mathcal{C}_\mathcal{T}$ and $(\mathcal{T}, \mathcal{C})$ behave the same on any input data graph.[1] Thus an infinite austere canonical model does not need to be built explicitly in order to perform validation. The rewriting method is interesting in its own right as it opens the way to reuse standard SHACL validators to perform validation in the presence of ontologies, and it thus joins the ranks of other rewriting-based methods for reasoning with infinite structures (see, e.g., [16, 17] and refereces therein).

2. Semantics of SHACL Validation with Ontologies

In this section, we describe in more detail the semantics we propose in [1] for validating SHACL shapes graphs in the presence of DL-Lite$_\mathcal{R}$ ontologies. More precisely, for a given ontology $\mathcal{T}$, data graph $\mathcal{A}$, and a shapes graph $(\mathcal{C}, \mathcal{G})$, where $\mathcal{C}$ is a set of constraints and $\mathcal{G}$ is a set of target atoms, we need to define when $\mathcal{A}$ validates $(\mathcal{C}, \mathcal{G})$ w.r.t $\mathcal{T}$. A natural first idea would be to follow the usual open-world semantics of ontologies and check for validation over all models of $\mathcal{A}$ and $\mathcal{T}$. While this works for positive constraints, it does not yield a natural result in the presence of negation. Consider a data graph $\mathcal{A}$ consisting of the facts $hasWingedPet(linda, blu)$, $Bird(blu)$, $PetOwner(linda)$, and an empty TBox $\mathcal{T}$. Let $(\mathcal{C}, \mathcal{G})$ be a shapes graph, where $\mathcal{C}$ only contains the constraint $s \leftarrow \exists hasWingedPet \land \neg\exists hasPet.Dog$ and the target to be checked for validation is $\mathcal{G} = \{s(linda)\}$. As the TBox is empty, we are in the usual setting of validation. Clearly, $\mathcal{A}$ validates $(\mathcal{C}, \mathcal{G})$ since $linda$ has a winged pet, and does not have a pet that is a dog. However, if we consider all possible models of $\mathcal{A}$ and $\mathcal{T}$, we have non-validation since there are models of $\mathcal{A}$ and $\mathcal{T}$ that include some other $hasPet$-fact for $linda$ and some pet b that is a Dog.

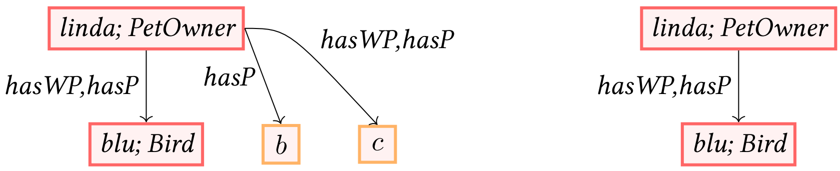
The problem is the *non-monotonicity* of SHACL, that is, adding facts to the data may cause a previously valid setting to become invalid. We want an intuitive semantics that coincides with the usual validation in case the ontology is empty. As done in related settings (see e.g., [12, 9, 13, 18]) we rely on the *chase* procedure [19] known from Knowledge Representation and Database Theory. Roughly speaking, a chase procedure takes as input a data graph and an ontology and iteratively applies the axioms of the ontology to the data by adding atoms over possibly fresh individuals until all the axioms in the ontology are satisfied. The result of the chase is a so-called *canonical* or *universal* model, and can be used as a representative of all the models. For DL-Lite$_\mathcal{R}$ ontologies, such chase procedures may not terminate and result in infinite models. There are several chase variants producing different canonical models [13]. While for positive constraints these differences do not matter, constraints with negation can distinguish between them, resulting in different validation answers, as illustrated in the example below.

The semantics we propose in [1] is based on a special chase procedure that constructs an *austere* canonical model. The main ingredient is an auxiliary notion of a *good successor*

[1] The impossibility of such a rewriting for SHACL with negation in Theorem 1 of [7] does not apply to our semantics, nor to the minimal-model semantics of [7], as acknowledged by the authors in personal communication.

configuration, which, for each object and its *type*, determines a set of successors that allows us to satisfy the axioms with as few fresh objects as possible while preserving the universality of the model. Our notion of austere canonical model is closely related to the *core* chase [13]. It will typically create fewer fresh successors than the *oblivious* chase, which, roughly speaking, applies the axioms of the ontology without first checking whether the axiom is already satisfied. It may also create fewer successors than the *restricted chase*, which may be sensitive to the order of rule applications. The semantics of validation with *DL-Lite*$_{\mathcal{R}}$ ontologies is given in terms of validation over the (possibly infinite) austere canonical model.

To illustrate the austere canonical model construction, consider the data graph $\mathcal{A}$ introduced above and the ontology $\mathcal{T}$ containing three axioms: (1)$PetOwner \sqsubseteq \exists hasPet$, (2) $hasWingedPet \sqsubseteq hasPet$, and (3) $PetOwner \sqsubseteq \exists hasWingedPet$. The good successor configuration will not generate a fresh successor for $linda$, since she has $blue$ as a winged pet, but also as a pet due to axiom (2). The austere canonical model (right of the figure) will only add a $hasPet$-role from $linda$ to $blue$. In contrast, the canonical model obtained from the oblivious chase (left of the figure) will introduce two fresh objects b, c, to satisfy the two existential axioms. In the figure, we use $hasP$ ($hasWP$) instead of $hasPet$ ($hasWingedPet$).



Now, consider the shapes graph $(\mathcal{C}, \mathcal{G})$ with $\mathcal{C} = \{s \leftarrow \exists hasPet.\neg Bird\}$ and $\mathcal{G} = \{s(linda)\}$. The shapes graph asks to validate whether $linda$ has a pet that is not a bird. Clearly, the austere canonical model provides the expected answer, as it does not validate $(\mathcal{C}, \mathcal{G})$. In contrast, the canonical model on the left-hand-side of the figure provides the unintended validation of $(\mathcal{C}, \mathcal{G})$.

3. Outlook

There are several directions for future work. In [1], we presented a rewriting algorithm for a restricted fragment of SHACL. Going forward, we plan to extend our approach to support more syntactic features of SHACL, like *complex path expressions* and *counting* (number restrictions on paths). We believe the mentioned features can be incorporated and supported by our rewriting approach in principle, but it requires a substantial extension. Another direction is to support ontology languages that go beyond OWL 2 QL. We believe our approach can be elegantly generalized to ontologies expressed in *Horn*-$\mathcal{SHIQ}$, but it is more challenging to support non-Horn ontology languages. An implementation of our approach also remains for future work. The rewriting algorithm was meant to demonstrate the principle feasibility of the approach. Our rewriting is best-case exponential; in particular, there is a rule (namely Rule 3 in Definition 5.3), which forces us to add exponentially many new constraints. A way to avoid this problem will be needed in order to achieve an efficient implementation of the rewriting. Extending the SHACL fragment to consider *unstratified* negation is also an interesting direction for future work.

Acknowledgments

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101034440.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. In addition, Šimkus was partially supported by the Austrian Science Fund (FWF) project P30873 and Ahmetaj was supported by the FWF and netidee SCIENCE project T1349-N.

References

[1] S. Ahmetaj, M. Ortiz, A. M. Oudshoorn, M. Simkus, Reconciling SHACL and ontologies: Semantics and validation via rewriting, 2023. ECAI 2023, to appear.

[2] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017. URL: http://www.cambridge.org/de/academic/subjects/computer-science/knowledge-management-databases-and-data-mining/introduction-description-logic?format=PB#17zVGeWD2TZUeu6s.97.

[3] W3C, Shape constraint language (SHACL), Technical Report. (2017). https://www.w3.org/TR/shacl.

[4] J. Corman, J. L. Reutter, O. Savkovic, Semantics and validation of recursive SHACL, in: D. Vrandecic, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L. Kaffee, E. Simperl (Eds.), The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I, volume 11136 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 318–336. URL: https://doi.org/10.1007/978-3-030-00671-6_19. doi:10.1007/978-3-030-00671-6_19.

[5] P. Pareti, G. Konstantinidis, F. Mogavero, Satisfiability and containment of recursive SHACL, Journal of Web Semantics 74 (2022) 100721. URL: https://www.sciencedirect.com/science/article/pii/S1570826822000130. doi:https://doi.org/10.1016/j.websem.2022.100721.

[6] M. Andresel, J. Corman, M. Ortiz, J. L. Reutter, O. Savkovic, M. Šimkus, Stable model semantics for recursive SHACL, in: Proc. of The Web Conference 2020, ACM, 2020, p. 1570–1580. doi:10.1145/3366423.3380229.

[7] O. Savkovic, E. Kharlamov, S. Lamparter, Validation of SHACL constraints over KGs with OWL 2 QL ontologies via rewriting, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. G. Gray, V. López, A. Haller, K. Hammar (Eds.), The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings, volume 11503 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 314–329. URL: https://doi.org/10.1007/978-3-030-21348-0_21. doi:10.1007/978-3-030-21348-0_21.

[8] M. Knorr, J. J. Alferes, P. Hitzler, Local closed world reasoning with description logics under the well-founded semantics, Artif. Intell. (2011). URL: http://dx.doi.org/10.1016/j.artint.2011.01.007. doi:10.1016/j.artint.2011.01.007.

[9] B. Motik, I. Horrocks, U. Sattler, Bridging the gap between OWL and relational databases,

in: Proceedings of WWW, ACM, 2007. URL: https://doi.org/10.1145/1242572.1242681. doi:10.1145/1242572.1242681.

[10] B. Motik, I. Horrocks, U. Sattler, Adding integrity constraints to OWL, in: C. Golbreich, A. Kalyanpur, B. Parsia (Eds.), Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, June 6-7, 2007, volume 258 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2007. URL: http://ceur-ws.org/Vol-258/paper11.pdf.

[11] W. Fan, F. Geerts, Relative information completeness, ACM Trans. Database Syst. (November 2010). URL: http://doi.acm.org/10.1145/1862919.1862924. doi:10.1145/1862919.1862924.

[12] S. Borgwardt, W. Forkel, Closed-world semantics for conjunctive queries with negation over ELH-bottom ontologies, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 6131–6135. URL: https://doi.org/10.24963/ijcai.2019/849. doi:10.24963/ijcai.2019/849.

[13] A. Deutsch, A. Nash, J. B. Remmel, The chase revisited, in: M. Lenzerini, D. Lembo (Eds.), Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada, ACM, 2008, pp. 149–158. URL: https://doi.org/10.1145/1376916.1376938. doi:10.1145/1376916.1376938.

[14] M. Krötzsch, OWL 2 profiles: An introduction to lightweight ontology languages, in: T. Eiter, T. Krennwallner (Eds.), Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, volume 7487 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 112–183. URL: https://doi.org/10.1007/978-3-642-33158-9_4. doi:10.1007/978-3-642-33158-9_4.

[15] K. R. Apt, H. A. Blair, A. Walker, Towards a Theory of Declarative Knowledge, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988, p. 89–148.

[16] T. Eiter, M. Ortiz, M. Simkus, T. Tran, G. Xiao, Query rewriting for Horn-SHIQ plus rules, in: J. Hoffmann, B. Selman (Eds.), Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada, AAAI Press, 2012. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4931.

[17] M. Bienvenu, M. Ortiz, Ontology-mediated query answering with data-tractable description logics, in: W. Faber, A. Paschke (Eds.), Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures, volume 9203 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 218–307. URL: https://doi.org/10.1007/978-3-319-21768-0_9. doi:10.1007/978-3-319-21768-0_9.

[18] S. Ellmauthaler, M. Krötzsch, S. Mennicke, Answering queries with negation over existential rules, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 5626–5633. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20503.

[19] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995. URL: http://webdam.inria.fr/Alice/.