Reasoning about Explanations for Non-validation in SHACL (Extended Abstract)*

Shqiponja Ahmetaj[2], Robert David[4], Magdalena Ortiz[1], Axel Polleres[2,3], Bojken Shehu[5], and Mantas Šimkus[1]

[1] Technical University of Vienna
[2] Vienna University of Economics and Business
[3] Complexity Science Hub Vienna, Austria
[4] Semantic Web Company
[5] Polytechnic University of Tirana

The Shape Constraint Language (SHACL) is a recently standardized language for expressing constraints on RDF graphs. It is the result of industrial and academic efforts to provide solutions for checking the quality of RDF graphs and for declaratively describing (parts of) their structure. We recommend [9] for an introduction to SHACL and its close relative *ShEx*. Among others, the SHACL standard provides a syntax for writing down constraints, as well as describes the way RDF graphs should be *validated* w.r.t. a given set of SHACL constraints. However, some aspects of validation were not completely specified in the standard, like the semantics of validation for constraints with cyclic dependencies. To address these shortcomings, recently several formalizations of SHACL have emerged, which describe it in logic-based languages with clear semantics. E.g., some works resort to first-order logic [6], while some use logic programming [2]. SHACL is closely related to expressive Description Logics (DLs). Such connections have already been observed in [10], where the authors reduce implication of SHACL constraints to concept subsumption in DLs. The key difference between SHACL and DLs is that SHACL makes the *closed-world assumption (CWA)*, while DLs use the *open-world assumption (OWA)*. To understand the difference, RDF graphs equipped with SHACL constraints can be thought of as DL knowledge bases in which all roles and some concept names are *closed predicates* in the context of DLs with closed predicates (see [8, 11]), i.e., where only some concept names are allowed to be non-closed predicates.

In SHACL, the basic computational problem is to check whether a given RDF graph G *validates* a SHACL document (C, T), where C is a specification of validation rules (*constraints*) and T is a specification of nodes to which the validation rules should apply (*targets*). In order to make SHACL truly useful and widely accepted, we need automated tools that implement not only validation, which results in "yes" or "no" answers, but also support the users in their efforts

* Partially supported by the Vienna Business Agency and the Austrian Science Fund (FWF) projects P30360 and P30873. Axel Polleres' work is supported by funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 957402.

to understand the reasons *why* a given graph validates or not against a given document. The SHACL specification stresses the importance of explaining validation outcomes and introduces the notion of *validation reports* for this purpose. If a graph validates a document, the standard has clear guidance how the validation reports should look like. However, the situation is different when the graph does not validate. The principles of validation reports in case of non-validation are left largely open in the standard, which specifies little beyond requiring that the node and constraint violated are indicated. It is not hard to see that, in general, there may be a very large number of possible reasons for a specific validation target to fail, and it is far from obvious what should be presented to the user in validation reports. This is precisely the topic of our study.

In this work[6], we advocate explanations in the style of *database repairs* [3] as one concrete way to provide explanations for the non-validation of SHACL constraints. This approach is closely related to subareas of KR&R like abductive reasoning, model-based diagnosis and counterfactuals, which have received very significant attention in the last decades and applied to a range of similar problems requiring explanatory services (see, e.g., [7, 12, 4, 5]).

The main goal of this work is to formalize the notion of explanations for SHACL, to define a collection of reasoning tasks for exploring explanations, and to characterize their computational complexity. In a nutshell, the contributions of this paper are as follows:

∘ To explain non-validation of a SHACL document (C, T) by an RDF graph G, we introduce the notion of a *SHACL Explanation Problem (SEP)*. A solution to a SEP is a pair (A, D) that describes a collection A of facts to be added to G and a collection D of facts to be deleted from G, so that the resulting graph does validate the document (C, T). We consider natural preference orders over explanations, and study also explanations that are minimal w.r.t. set inclusion or w.r.t. cardinality. We illustrate the use of explanations with some examples.

∘ We define a collection of inference services for reasoning about explanations for non-validation. We start with the basic tasks of recognizing whether a given candidate is indeed a (preferred) explanation, and deciding whether a (preferred) explanation exists. We also define the problems of checking whether a given atom is relevant (resp., necessary) as an addition or as a deletion in some explanation (resp., all explanations). These tasks are reminiscent of basic reasoning problems in logic-based abduction [7].

∘ We study the computational complexity of the introduced reasoning tasks and characterize both combined and data complexity. Our results range from tractability to completeness for the second level of the polynomial hierarchy.

∘ After studying the general setting, we turn our attention to *non-recursive* SHACL constraints. We show that with one exception, reasoning about explanations in the presence of non-recursive constraints does not become easier in terms of computational complexity. The exception is the problem of recognizing an explanation, which becomes tractable in the absence of a preference order.

[6] This is an extended abstract of [1].

As a side result we show that SHACL validation in the presence of non-recursive constraints is P-complete.

∘ Finally we consider a generalization of SEPs with *restricted explanation signatures*. This useful feature allows, e.g., to specify that some classes and properties are *read-only*, prohibiting deletions from them during explanations. Also for this setting, we establish a collection of complexity results, including the case of non-recursive constraints.

References

1. Ahmetaj, S., David, R., Ortiz, M., Polleres, A., Shehu, B., Šimkus, M.: Reasoning about explanations for non-validation in SHACL. In: Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021. To appear.
2. Andresel, M., Corman, J., Ortiz, M., Reutter, J.L., Savkovic, O., Šimkus, M.: Stable model semantics for recursive SHACL. In: Proc. of The Web Conference 2020. p. 1570–1580. WWW '20, ACM (2020). https://doi.org/10.1145/3366423.3380229, https://doi.org/10.1145/3366423.3380229
3. Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent query answers in inconsistent databases. In: Proc. of PODS. pp. 68–79. ACM Press (1999). https://doi.org/10.1145/303976.303983, https://doi.org/10.1145/303976.303983
4. Calvanese, D., Ortiz, M., Simkus, M., Stefanoni, G.: Reasoning about explanations for negative query answers in DL-Lite. J. Artif. Intell. Res. **48**, 635–669 (2013). https://doi.org/10.1613/jair.3870, https://doi.org/10.1613/jair.3870
5. Ceylan, İ.İ., Lukasiewicz, T., Malizia, E., Molinaro, C., Vaicenavicius, A.: Explanations for negative query answers under existential rules. In: Proc. of KR 2020. pp. 223–232 (2020). https://doi.org/10.24963/kr.2020/23, https://doi.org/10.24963/kr.2020/23
6. Corman, J., Reutter, J.L., Savkovic, O.: Semantics and validation of recursive SHACL. In: Proc. of ISWC'18. Springer (2018). https://doi.org/10.1007/978-3-030-00671-6_19, https://doi.org/10.1007/978-3-030-00671-6_19
7. Eiter, T., Gottlob, G.: The complexity of logic-based abduction. J. ACM **42**(1), 3–42 (1995). https://doi.org/10.1145/200836.200838, https://doi.org/10.1145/200836.200838
8. Franconi, E., Ibáñez-García, Y.A., Seylan, I.: Query answering with DBoxes is hard. Electr. Notes Theor. Comput. Sci. (2011). https://doi.org/10.1016/j.entcs.2011.10.007, http://dx.doi.org/10.1016/j.entcs.2011.10.007
9. Gayo, J.E.L., Prud'hommeaux, E., Boneva, I., Kontokostas, D.: Validating RDF Data. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers (2017). https://doi.org/10.2200/S00786ED1V01Y201707WBE016, https://doi.org/10.2200/S00786ED1V01Y201707WBE016
10. Leinberger, M., Seifer, P., Rienstra, T., Lämmel, R., Staab, S.: Deciding SHACL shape containment through description logics reasoning. In: Proc. of ISWC 2020. Lecture Notes in Computer Science, vol. 12506, pp. 366–383. Springer (2020). https://doi.org/10.1007/978-3-030-62419-4_21, https://doi.org/10.1007/978-3-030-62419-4_21

11. Lutz, C., Seylan, I., Wolter, F.: Ontology-based data access with closed predicates is inherently intractable(sometimes). IJCAI/AAAI (2013), http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6870
12. Van Harmelen, F., Lifschitz, V., Porter, B.: Handbook of knowledge representation. Elsevier (2008)