SHACL Constraint Validation during SPARQL Query Processing

Philipp D. Rohde
supervised by Maria-Esther Vidal
Leibniz University of Hannover
TIB Leibniz Information Centre for Science and Technology
Hannover, Germany
philipp.rohde@tib.eu

ABSTRACT

The importance of knowledge graphs is increasing. Due to their application in more and more real-world use-cases the data quality issue has to be addressed. The Shapes Constraint Language (SHACL) is the W3C recommendation language for defining integrity constraints over knowledge graphs expressed in the Resource Description Framework (RDF). Annotating SPARQL query results with metadata from the SHACL validation provides a better understanding of the knowledge graph and its data quality. We propose a query engine that is able to efficiently evaluate which instances in the knowledge graph fulfill the requirements from the SHACL shape schema and annotate the SPARQL query result with this metadata. Hence, adding the dimension of explainability to SPARQL query processing. Our preliminary analysis shows that the proposed optimizations performed for SHACL validation during SPARQL query processing increase the performance compared to a naive approach. However, in some queries the naive approach outperforms the optimizations. This shows that more work needs to be done in this topic to fully comprehend all impacting factors and to identify the amount of overhead added to the query execution.

1 INTRODUCTION

Knowledge graphs still experience an exponential growth [11] and became expressive data structures that provide a unified view of a multitude of data sources. Knowledge graphs are used in large IT companies [13] as well as for domain-specific data like in biomedicine [12]. These use-cases prove the potential of knowledge graphs but also show the need for efficient means of knowledge graph creation, curation, and understanding.

The Shapes Constraint Language (SHACL)[1] is the W3C recommendation language for declaratively defining integrity constraints over data expressed in the Resource Description Framework (RDF)[2]. SHACL models integrity constraints as a network of shapes, called *shape schema*. A shape consists of all constraints against attributes of a specific RDF target resource. Requirements on properties associating two targets are modeled as links between the shapes. SHACL is being used in real-world scenarios, and it is also adopted in industrial consortia, e.g., the International Data Space (IDS) [5], to represent integrity constraints in the reference architectures. While SHACL exhibits clarity and readability, SHACL shape schema validation can face tractability issues. In general, the problem is intractable [8]. State-of-the-art approaches have identified tractable fragments which cover most real-world use-cases and provided means for the efficient validation of those fragments [6, 9].

SPARQL Protocol And RDF Query Language (SPARQL)[3] is the W3C recommendation language to query RDF data. Much work has been done by the community to study different approaches to improve the query performance of SPARQL [2]. Recent work [1, 14] also considers the integrity constraint schema for cost-based approaches to query optimization. However, many publicly accessible knowledge graphs suffer from quality issues even if they are curated [11]. This work aims at increasing the explainability of SPARQL query results by annotating it with the result from the SHACL shape schema validation while minimizing execution time to evaluate which instances are valid.

2 MOTIVATION

Consider an RDF knowledge graph containing data about universities from the LUBM [10] benchmark. Given this data, the SPARQL query in Figure 1a retrieves the names of full professors that have an email address and work at Department0 of University1, their research interest, and the URI of the university they got their PhD from. Figure 1b depicts the integrity constraints defined for professors, departments, and universities. Instances of all three classes have to have exactly one name. Departments are a sub-organization of at least one university. Professors additionally have at least one email address and research interest. Furthermore, professors work for at least one department and obtained their PhD from at least one university. An instance associated with the professor shape is valid if the instance meets all requirements, i.e., the instance fulfills the *intra-shape constraints*, i.e., the constraints not linking to other shapes, as well as the *inter-shape constraints*, i.e., the constraints linking to other shapes. An instance meets the inter-shape constraints if the instances linked to fulfill all requirements inflicted on them. The result of the query in Figure 1a is presented in Figure 1c. Note that in the example data only eight of 1,000 universities have a name. Hence, only three of the seven query results meet all integrity constraints defined for the data. That does not falsify the result reported. However, it provides an explanation for why the query has only three answers when asking for the university name as well; by adding the triple pattern `?uni ub:name ?uname` to the query. Adding metadata, e.g., from the validation of a SHACL shape schema, to the result of a SPARQL query helps in understanding the answers retrieved from an RDF knowledge graph.

Proceedings of the VLDB 2021 PhD Workshop, August 16th, 2021. Copenhagen, Denmark.

[1] https://www.w3.org/TR/2017/REC-shacl-20170720/
[2] https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
[3] https://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

```
PREFIX ub: <http://swat.cse.lehigh.edu/onto/univ-bench.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?name ?ri ?uni WHERE {
  ?prof  rdf:type  ub:FullProfessor ;
     ub:name  ?name ;
     ub:worksFor  <http://www.Department0.University1.edu> ;
     ub:doctoralDegreeFrom  ?uni ;
     ub:emailAddress  ?email ;
     ub:researchInterest  ?ri .
} ORDER BY ?prof
```

(a) SPARQL Query

(b) SHACL Shape Schema

name	ri	uni	__meta__
FullProfessor0	Research6	http://www.University6.edu	all requirements met
FullProfessor1	Research23	http://www.University7.edu	all requirements met
FullProfessor2	Research15	http://www.University1.edu	all requirements met
FullProfessor3	Research10	http://www.University888.edu	University888 violates name constraint
FullProfessor4	Research19	http://www.University358.edu	University358 violates name constraint
FullProfessor5	Research8	http://www.University996.edu	University996 violates name constraint
FullProfessor6	Research12	http://www.University87.edu	University87 violates name constraint

(c) Query Result with Annotations from Quality Assessment

Figure 1: Motivating Example over an RDF University System (LUBM Benchmark). (a) A SPARQL query retrieving the names of the *FullProfessors* working for dept0 of uni1, their research interest, and the URI of the university they got their PhD from. (b) The SHACL shape schema used for validating the data. (c) The result of the query in (a) annotated with the results from the SHACL validation of the shape schema in (b). Three out of seven query results meet all the constraints. The other four results include universities violating the constraint on the predicate *ub:name*, i.e., they have no or more than one name in the data.

3 RELATED WORK

SHACL Validation. Due to the increasing importance of RDF data, the need for efficiently checking integrity constraints over RDF data emerge. Corman et al. [8] propose a new semantics for recursive SHACL since the semantics of recursions are left open in the SHACL specification. Based on this work, they identify three tractable fragments of SHACL [7] and develop an algorithm able to validate SHACL shape schemas of these fragements [6]. Andreşel et al. [4] propose to use stable models known from Answer Set Programming (ASP) to validate SHACL shape schemas. This results in an even stricter semantics for recursive SHACL. This approach allows to translate the validation into a logic progamm that can be solved using of-the-shelf ASP solvers. In contrast to these logic approaches to the validation of a SHACL shape schema, Figuera et al. [9] propose Trav-SHACL which aims at improving the incremental behavior and scalability of the validation problem. They make use of the fragments and algorithms identified by Corman et al. [6, 7] and improve the performance by interleaving the data retrieval, rule grounding, and saturation steps. Additionally, Trav-SHACL optimizes the queries sent to the endpoint in order to identify invalid entities as fast as possible. While this work is dependent on efficient validation of SHACL, it is not the main focus.

Query Processing. Recently integrity constraints over RDF data – expressed either in SHACL or ShEx [15] – have been included in studies about optimization of SPARQL queries. Abbas et al. [1] propose rules for well-formed ShEx schemas and SPARQL query optimization by triple pattern reordering based on a well-formed ShEx schema. The triple patterns are ranked based on the hierarchical structure of the shapes within the ShEx schema, i.e., shape inclusion, as well as the predicate distribution, i.e., the generality of the predicate. Shapes that are being included in other shapes get ranked higher as they are assumed to be more selective due to the *well-formation cardinality rule*. This rule forces m-to-n relations where $m > n$ to be defined on the m-side. Triple patterns with predicates that are unique for one shape are ranked higher than the ones with more general predicates as they are assumed to be more selective. Rabbani et al. [14] propose an extension of SHACL including statistics into the definition of a SHACL shape. These statistics capture the total triple count, minimum and maximum number of triples for each instance, and the number of distinct objects. The proposed query optimizer uses the information encoded in the SHACL shapes for cardinality estimation as part of a cost-based query planner. The authors show that the approach generates query plans that are cheaper or at most of the same cost as state-of-the-art heuristic query planners. Additionally, their approach requires considerably less time and space for the pre-processing compared to other cost-based approaches which rely on the existence of hard-to-compute statistics. These optimization techniques are sound, but they assume that the data complies to the integrity constraints. However, especially in publicly available RDF data sources, this assumption does not hold. On the contrary, most of the data suffers from low quality, e.g., missing values. Our approach aims at explaining the query result based on the validation of the integrity constraints, i.e., annotating the query result.

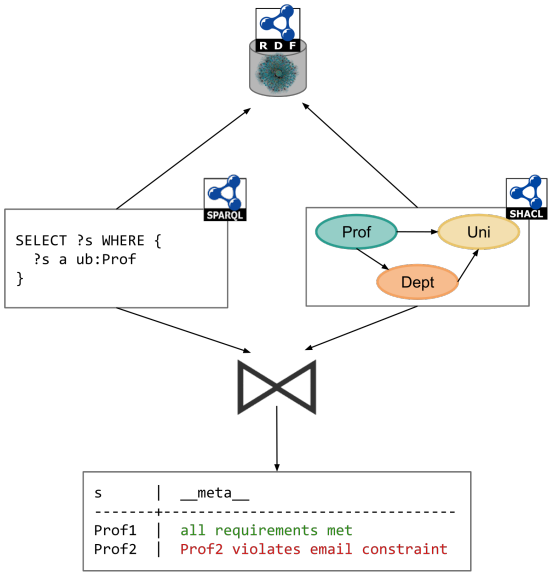

Figure 2: Approach Overview. A SPARQL query is executed over an RDF knowledge graph. A SHACL shape schema is validated against the same knowledge graph. The result of the query and the validation are joined to form the annotated query result improving explainability.

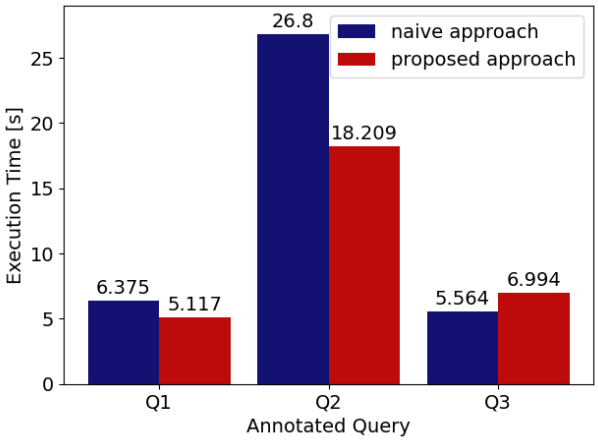

Figure 3: Preliminary Results over WatDiv. *Naive approach* is evaluating the complete shape schema while *proposed approach* implements the proposed optimizations. For queries Q1 and Q2 the optimizations increase the performance. However, for Q3 the naive approach performs better.

4 PROBLEM DEFINITION

Problem Statement. Given an RDF graph $\mathcal{G} = \langle V_{\mathcal{G}}, E_{\mathcal{G}} \rangle$, a SHACL shape schema $\mathcal{S} = \langle S, \text{TARG}, \text{DEF} \rangle$, and a SPARQL query Q, the problem of annotating the SPARQL query result $[[Q]]^{\mathcal{G}}$ with the SHACL validation result $[\mathcal{S}]^{\mathcal{G}}$ is to match the instances in $[[Q]]^{\mathcal{G}}$ with the instances in $[\mathcal{S}]^{\mathcal{G}}$ such that the execution time required for the annotation of the query result, i.e., evaluating the shape schema and matching it against the query result, is minimized.

Solution. The work necessary to address the problem of enriching SPARQL query results with metadata to increase the explainability can be broken down into the following dimensions:

i) Query Decomposition: SPARQL queries can be decomposed into subject star-shaped sub-queries [17]. All triple patterns of such a sub-query have their subject in common with the other triple patterns. It is likely that the instances in the data that match the star-shaped sub-query represent one class of instances.

ii) SHACL Validation: A crucial part of this work is the efficient validation of SHACL shape schemas since either the complete RDF knowledge graph or an appropriate subset has to be validated. Interleaving the different steps of the SHACL validation and optimizing the queries sent to the SPARQL endpoint improve the performance [9]. However, in this use-case more optimizations can be done. This is due to the fact that most queries will only include a subset of the SHACL shapes present in the shape schema. Shapes in the shape schema that do not play a role in the validation result of the shapes covered by the query unnecessarily consume resources and can be omitted during the validation. For the query and SHACL shape schema in Figure 1 no shapes can be removed from the validation since the query targets the Professor shape which is linked to all other shapes via inter-shape constraints.

iii) SPARQL Query Result Annotation: For a SHACL shape it is common to impose integrity constraints on a set of instances that share the same properties. Hence, it makes sense to decompose the original SPARQL query into subject star-shaped sub-queries and annotate the sub-queries first. Figure 2 shows the approach to annotate an exemplary sub-query with the result from the SHACL validation based on the motivating example (see Figure 1). The complete shape schema might also include requirements for students and research assistants. Once all the sub-queries have been executed and annotated, the SPARQL operators, like join and union, have to combine the results from each sub-query. Part of this work would be the formalization of the semantics and SPARQL operators in presence of annotations from the SHACL validation.

5 RESULTS SO FAR

So far, we studied the efficient validation of SHACL shape schemas. In [9] we reported our work on the topic. We proposed new optimization techniques to increase the performance and continuous behavior of SHACL shape schema validation. Furthermore, we started to investigate the impact of annotating the result of star-shaped SPARQL queries with the result from the aforementioned validation. In order to annotate the query result, we modified the XJoin [16] operator to match instances from the query result with the instances from the SHACL validation. In this process, we identified several factors that impact the performance of such a system. To increase the performance, we defined further optimization techniques that can be applied in case the annotation of the query result is needed. The results of a preliminary analysis over WatDiv [3] are shown in Figure 3. All queries reported are subject star-shaped queries created from the original WatDiv queries. The queries are highly selective and cover the classes *Role* and *ProductCategory*. For two of the three queries, the proposed optimizations increase the performance. However, for the third query, the optimizations lead to a worse performance. This needs to be studied further.

6 RESEARCH PLAN

This work is based on previous work in SHACL validation as well as SPARQL query processing. The goal is to create a SPARQL query engine with explainable results through annotations from the validation of a SHACL shape schema associated with the data.

SHACL Validation. As a first step, we started to investigate the problem of efficiently validating SHACL shape schemas (see Section 5). Based on the tractable fragments of SHACL identified by Corman et al. [7] we proposed Trav-SHACL [9], a SHACL validator that interleaves the data collection, rule grounding, and saturation steps. Additionally, Trav-SHACL rewrites the SPARQL queries sent to the endpoint in order to optimize the queries in terms of execution time. We found that interleaving the stages of the validation process and optimizing the queries improves the performance. Trav-SHACL benefits from scenarios with data of low quality as it is designed to find invalid instances fast; by skipping the evaluation of further integrity constraints of already invalid instances which does comply with the SHACL specification. This behavior might need to be turned off in the case a complete (detailed) explanation is required.

SPARQL Query Result Annotation. First, we plan to annotate the query result of subject star-shaped queries, i.e., SPARQL queries where all triple patterns of the query have the same subject variable. As described in Section 5, we already made some progress on that. In order to reduce the overall execution time, the validation time of the SHACL shape schema needs to be reduced as this is the main bottleneck in this approach. In order to achieve that, we defined optimization techniques to limit the number of integrity constraints and instances that need to be checked during the SHACL validation. Our current results show that this approach can be done. The next step is to formalize the optimization techniques we came up with so far. Afterwards, we plan to extend the SPARQL query result annotation to queries containing any basic graph pattern (BGP) based on the decomposition of the BGP into star-shaped subqueries (SSQ) [17]. This requires us to formalize an extension of the SPARQL algebra to include and aggregate the annotation of SSQs in the SPARQL operators, e.g., join and union. The ultimate goal is to have a SPARQL query engine that is able to provide explainable query results by adding metadata from the quality assessment – in our case SHACL validation – to the individual results of the query.

Additional Annotations. Adding information about the data quality, e.g., from the validation of a SHACL shape schema associated with the data, is only a first step towards explainability of SPARQL query results. Imagine a recruitment system linked to multiple SPARQL endpoints. Some of the data sources might be private knowledge graphs containing personal information, e.g., the degrees obtained from a university. In the process of reviewing a job application, a recruiter gets read access to the personal information of the applicant. Currently, the recruiter does not know if the data retrieved is actually true. Now assume that the system uses a blockchain – or similar technology – to record all transactions, e.g., adding RDF triples to a knowledge graph. In this case, it is possible to annotate the query result with the information of who actually added each triple. Back to the recruiter, the systems also shows the name of the university as the entity who added the information about the applicant's degree, so the chances are high that the applicant really holds the degree.

7 CONCLUSIONS

RDF knowledge graphs are gaining momentum. Therefore, the issue of data quality must be addressed. SHACL is the W3C recommendation language for integrity constraints over RDF. SHACL is used in more and more use-cases. Hence, the efficient validation of SHACL shape schemas is necessary, so that SHACL is able to scale up to real-world scenarios with big data. Explainability and bias detection are hot topics at the moment. Enriching SPARQL query results with annotations from the SHACL shape schema validation is a first step towards explainable query results.

ACKNOWLEDGMENTS

This work has been partially supported by the EU H2020 RIA funded projects QualiChain (No 822404) and CLARIFY (No 875160), and the ERAMed project P4-LUCAT (No 53000015).

REFERENCES

[1] Abdullah Abbas, Pierre Genevès, Cécile Roison, and Nabil Laya"ida. 2018. Selectivity Estimation for SPARQL Triple Patterns with Shape Expressions. In *Web Engineering. ICWE 2018.*

[2] Waqas Ali, Mohammad Saleem, Bin Yao, Aidan Hogan, and Axel-Cyrille Ngonga Ngomo. 2021. A Survey of RDF Stores & SPARQL Engines for Querying Knowledge Graphs. *CoRR* abs/2102.13027 (2021).

[3] Güneş Aluç, Olaf Hartig, M. Tamer Özsu, and Khuzaima Daudjee. 2014. Diversified Stress Testing of RDF Data Management Systems. In *The Semantic Web – ISWC 2014.*

[4] Medina Andreşel, Julien Corman, Magdalena Ortiz, Juan L. Reutter, Ognjen Savković, and Mantas Šimkus. 2020. Stable Model Semantics for Recursive SHACL. In *ACM – The Web Conference.*

[5] Sebastian R. Bader, Jaroslav Pullmann, Christian Mader, Sebastian Tramp, Christoph Quix, Andreas W. Müller, Haydar Akyürek, Matthias Böckmann, Benedikt T. Imbusch, Johannes Lipp, Sandra Geisler, and Christoph Lange. 2020. The International Data Spaces Information Model - An Ontology for Sovereign Exchange of Digital Content. In *International Semantic Web Conference ISWC.*

[6] Julien Corman, Fernando Florenzano, Juan L. Reutter, and Ognjen Savković. 2019. SHACL2SPARQL: Validating a SPARQL Endpoint against Recursive SHACL Constraints. In *International Semantic Web Conference ISWC Satellite Events.*

[7] Julien Corman, Fernando Florenzano, Juan L. Reutter, and Ognjen Savković. 2019. Validating SHACL Constraints over a SPARQL Endpoint. In *International Semantic Web Conference ISWC.*

[8] Julien Corman, Juan L. Reutter, and Ognjen Savković. 2018. Semantics and Validation of Recursive SHACL. In *International Semantic Web Conference ISWC.*

[9] Mónica Figuera, Philipp D. Rohde, and Maria-Esther Vidal. 2021. Trav-SHACL: Efficiently Validating Networks of SHACL Constraints. In *ACM – The Web Conference.*

[10] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. 2005. LUBM: A Benchmark for OWL Knowledge Base Systems. *Web Semantics* 3, 2–3 (2005).

[11] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2020. Knowledge Graphs. *CoRR* abs/2003.02320 (2020).

[12] David N. Nicholson and Casey S. Greene. 2020. Constructing knowledge graphs and their biomedical applications. *Comput Struct Biotechnol J.* 18 (2020).

[13] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* 62, 8 (2019).

[14] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. 2021. Optimizing SPARQL Queries using Shape Statistics. In *Advances in Database Technology – EDBT 2021.*

[15] Katherine Thornton, Harold Solbrig, Gregory S. Stupp, Jose Emilio Labra Gayo, Daniel Mietchen, Eric Prud'hommeaux, and Andra Waagmeester. 2019. Using Shape Expressions (ShEx) to Share RDF Data Models and to Guide Curation with Rigorous Validation. In *The Semantic Web – ESWC 2019.*

[16] Tolga Urhan and Micheal J. Franklin. 2000. XJoin: A Reactively-Scheduled Pipelined Join Operator. *IEEE Data Eng. Bull.* 23, 2 (6 2000).

[17] Maria-Esther Vidal, Edna Ruckhaus, Tomas Lampo, Amadís Martínez, Javier Sierra, and Axel Polleres. 2010. Efficiently Joining Group Patterns in SPARQL Queries. In *The Semantic Web: Research and Applications. ESWC 2010.*