Knowledge Graph Repair

Robert David[1,2][0000−0002−3244−5341]

Institute for Data, Process and Knowledge Management,
Vienna University of Economics and Business, Austria

Abstract. The Semantic Web provides standards for knowledge graphs (KGs), which have become popular for solving data heterogeneity problems in enterprises, since they allow for flexible data modelling and integration via linking of graphs. This flexibility requires standards to ensure data quality and consistent state, such as the Shapes Constraint Language SHACL. However, SHACL does not provide the means to explain why constraint violations occur and how the KG can be repaired to conform to the constraints. Also, repairs for a KG can come with a high number of different alternative choices to pick from, where we need a way to determine preferences in practice. Finally, knowledge in the KG itself can be exploited for repairs to determine fresh values and preferred choices and to identify incorrect data from a real-world perspective. For these challenges, we aim to develop a system that combines logic-based repairs and data-driven analysis for a repair approach that concludes KGs towards a quality fix point. The approach will not only be defined at the formal level, but we also will provide prototypical implementations for practical experiments, thereby positioning it at the intersection of theoretical and applied research. Use cases shall be provided from companies, projects and open data to better understand how repairs can be applied effectively in practice. With this work we contribute to improving the quality of KGs by providing intelligent knowledge graph repair.

Keywords: Knowledge Graphs · Shapes Constraint Language · SHACL · Data Repair · Data Quality · Logic Programming · Data-Driven Anayl-sis · Hybrid AI

1 Motivation

Knowledge graphs (KGs) [13] are often used in enterprises for consolidation of heterogeneous data sources. The flexible approach of modelling graphs using the Resource Description Framework (RDF) makes it possible to link together various data sources and thereby integrate them into an enterprise knowledge graph (EKG) [11]. However, problems like inconsistencies and incomplete data can be introduced. These problems generally represent quality issues that need appropriate mechanisms to detect and manage them in practice. This generally applies to KGs, but it is especially important in an enterprise scenario, where business decisions depend on the consolidated information.

The Semantic Web traditionally uses open world reasoning based on RDFS and OWL [3], which has very limited use regarding the detection of inconsistencies. While OWL can express restrictions, an OWL reasoner uses them for classification and not as constraints on the data. The later introduced Shapes Constraint Language SHACL [15] can detect data inconsistencies based on constraints and return a report about violations. However, these reports do not explain why a violation has happened and they do not provide sufficient information on how to correct the data. In a data consolidation scenario, we mostly consider changing the KG itself without changing the underlying schema. Knowledge graph refinement, as discussed in [18], focuses on adding missing knowledge to the graph and identifying wrong information in the graph, and thereby increasing the quality of the graph regarding the schema. As the quality of graph data is of high importance in practice, having a system which can determine such refinements so that constraint violations can be repaired would be of high value.

In the following, we present the state of the art for graph standards, ontology and database repairs and data-driven analysis. Then we continue with formulating research questions and contributions our work will provide. In the research methodology and approach section, we show how we plan to incrementally address the questions, followed by the evaluation plan. We then provide a report on the intermediary results up to now. Finally, we present conclusions and next steps.

2 State of the Art

The state of the art related to this thesis covers knowledge graph technologies, which are the foundation of our work. We also look into approaches to repair data for ontologies and relational databases and the complexity of the repair problem. Finally, we need to understand the broad field of data-driven analysis methods, which can be used to determine repair choices.

2.1 Knowledge Graphs and the Semantic Web

The Semantic Web [16] is a set of standards for knowledge representation for the world wide web. It includes languages to describe the semantics of (graph) data, RDF Schema RDFS and the Web Ontology Language OWL [3]. These languages allow for reasoning about the data in the form of logical conclusions. Because links to graphs can be created by anyone on the web, these conclusions use the open world assumption for monotonic reasoning, but thereby accepting problems introduced by contradicting data.

Enterprise Knowledge Graphs Originally created for open data publishing, the Semantic Web standards were discovered to be sufficiently flexible and expressive to be used in enterprise contexts to solve data integration problems under the term Enterprise Knowledge Graphs (EKG) [11]. On the organisational side,

EKGs differ from public knowledge graphs usually by being limited to the enterprise's data sources and the need for high quality to avoid false information which might become problematic from a business point of view. Therefore, the quality of graph datasets needs to be defined in a principled manner in order to be able to detect conflicts and possibly repair them [7].

Graph Constraint Languages There are currently two constraint languages for knowledge graphs. The shapes constraint language SHACL [15] is the W3C recommendation for constraint validation in the Semantic Web. The shape expressions language ShEx [19] is developed in a W3C community group and publicly available. SHACL and ShEx are both based on the idea of shapes that group together constraints to be evaluated on nodes in the knowledge graph. They differ in syntax, and in some details regarding constraint validation and reporting, and they define different semantics for recursion. Semantics for shape validation are discussed in general in the scientific community regarding what makes sense for data quality. Generally, the provided information should assist in fixing the data to achieve conformance with the constraints and thereby improve the quality of the graph data.

2.2 Repair Approaches

Relational Database Repairs Inconsistent data, as a consequence of violating integrity constraints, is well-researched for relational databases. Two strategies for coping with inconsistent relational data are presented in [5]. First, consistent query answering (CQA) does not resolve the inconsistencies but provides answers to queries based on a consistent subset of the data. Second, data cleaning repairs the inconsistencies while trying to preserve as much information as possible by making minimal changes. This strategy can result in different choices on how to change the data. One way of specifying and implementing repairs is to represent them as logic programs, e.g. using disjunctive Datalog with stable model semantics, which is described in [5]. These programs modify a database to achieve conformance with a set of integrity constraints.

Ontology Repairs Related to our work are also reasoning tasks for explanations and repairs for ontologies, often in the context of Description Logics. Debugging and repair of OWL ontologies [14] addresses logical contradictions in OWL-DL ontologies and proposes a framework to repair such defects. Examples of work on explanations for negative query answers can be found in [8] and [9]. A recent work on Description Logic ontologies is [4], which addresses optimal repairs in the scenario where the schema (TBox) is assumed to be correct, while the data (ABox) needs to be repaired. Optimal repairs in this context are repairs which preserve as much as possible from the logical consequences of the ontology. Similar, an approach to repair EL ontologies is presented in [17] using a combination (or trade-off) of weakening and completing to mitigate the negative effects of removing wrong axioms. Finally, the relation between Description Logics and

SHACL is discussed in [6]. The idea is to bridge the two communities and provide work about Description Logic as a formal foundation for future SHACL developments.

Rule-based Repairs Rule-driven inconsistency resolution is described in [12], which assumes rule-generated descriptions of entities in KGs by using ontologies. In this scenario, quality problems can be introduced when ontology terms are used without alignment with the rules, thereby leading to violations of restrictions defined by the ontologies. A rule-driven methodology is introduced which determines refinements which should be applied to the rules and the ontology. Experts still need to decide manually if the rules or the ontology should be changed.

2.3 Data-driven Analysis

Some aspects of repairs cannot always be covered by formal constraints alone. A source of repair information is the data graph itself, which can be analysed to decide on fresh values, pick repair choices and identify false data from a real-world perspective beyond formal constraints. The field of concluding new knowledge from sample data is very broad and there are many different approaches to do so, with the most prominent being machine learning methods like Graph Neural Networks [20]. The field of Knowledge Graph Completion [10] focuses on predicting graph structures and values based on statistical analyses of existing graph data. It contains many different approaches, which range from probabilistic methods to modern machine learning techniques, like deep learning and large language models. Such data-driven analyses, when combined with crisp semantic descriptions from the data set, can be a source of practically viable repair options.

3 Problem Statement and Contributions

In the course of the thesis, we plan to address the following research questions and develop solutions for knowledge graph repair. On the theoretical side we provide novel formalisms to analyse graphs regarding repairs based on constraints. On the practical side, we apply prototypes to practical scenarios to gain further insights for practically viable repairs. With this work we aim to contribute to the advance of the academic discourse on knowledge graph quality. In the following, we identify 3 main research questions.

3.1 Explaining and repairing Constraint Violations

The first main research question is about how to define and implement repairs for graph data based on constraints. This is an open question in the research community that was not yet addressed. To limit the scope, our work focuses on SHACL as a constraint language. In the following, we provide a motivating example for the repair problem.

Example 1. Assume a shapes graph with a shape StudentShape (left) and a data graph (right), written in Turtle syntax:

```
:StudentShape a sh:NodeShape;
    sh:targetNode :Ben;
    sh:property [
        sh:path :studentID;
        sh:maxCount 1;
    ].
```

```
:Ben :studentID "2119110",
                "1716110".
```

The StudentShape defines that a student must not have more than one student ID. In this example, we can see that *Ben* has two student IDs. *Ben* is a target node for StudentShape and violates the *sh:maxCount* constraint.

SHACL is designed to identify quality problems by specifying constraints grouped into shapes and then checking if the graph data satisfies them. The process of validation returns reports for constraint violations. However, reports do not provide an explanation for why a constraint is violated and how this can be solved. In example 1, we could delete either one of the student IDs to make *Ben* validate StudentShape, so there are two (minimal) repair choices. Providing automatic solutions for violations in more complex scenarios is not trivial. Explanations are non-deterministic and have to be determined globally for a data graph. Currently, it is open how to compute such repairs for knowledge graphs.

To develop a solution for repairs, we first need a formal definition for explanations to represent why these non-validations of SHACL constraints happen. These explanations are a preliminary step and the basis to determine how to change the data graph to achieve conformance with the constraints. We also discuss which kinds of explanations we would like to have from a practical point of view. For example, we can argue for minimality of changes, because we want to preserve as much of the original data graph as possible. Second, we investigate how to transfer existing repair approaches for relational databases to knowledge graphs and how to combine them with SHACL.

RQ1: How can we compute repairs that correct graph data to conform to SHACL constraints and how can this be implemented?

- *RQ1.1: What is our understanding of an explanation for SHACL constraint violations?*
- *RQ1.2: How can we compute SHACL repairs based on explanations and provide an implementation?*
- *RQ1.3: How well does the repair approach scale for practical purposes?*

3.2 Repair Strategies

The second main research question continues from the first main research question and asks about ways to represent user preferences for the repairs on top of what SHACL repairs can express. The motivation for this question comes from two open issues which are not addressed by the first question.

- First, users might decide for certain elements of the data graph that they should not be added or deleted by the repairs, which basically means that they are read-only. In a less restrictive situation the user might still allow these elements to be added or deleted, but it would not be a preferred situation to pick them if there are alternative choices to repair the data graph.
- Second, in the situation where there is a high number of possible and equally optimal repair choices, users can decide on preferences to reduce the number of choices and thereby make it easier to pick one choice.

Technically, preferences can be addressed by weighing different repair choices and determining the optimally weighted repair. We will apply this approach to extend the basic SHACL repair implementation.

RQ2: Based on our definition of repairs, how can we provide formalized repair strategies so that users can define optimal repair choices for specific use cases.

3.3 Data-driven repairs choices

The third main research question again continues from the previous main research questions and brings in data-driven methods to SHACL repairs for determining repair preferences. The idea is to create a hybrid system of formal and data-driven approaches.

This question is motivated at first by the question of how to come up with fresh values if a SHACL repair needs the addition of data. Such values usually cannot be determined by looking at the constraints and need to come from a different source. An approach is to take into account information from the (existing) data graph that can provide insights into how to choose values. Also, we can determine which choices to preferably pick for repairs. Finally, we can determine outliers which might be incorrect data. As described, data-driven analysis to determine such values is a broad and well-researched field. In our work, we focus on the integration of such methods for repair strategies and will not research new data-driven methods.

RQ3: How can we formally integrate data-driven methods into repair strategies to conclude missing values, detect outliers and pick from multiple choices?

3.4 Future Questions

Finally, we would like to point out other interesting research questions, which were identified, but will not be addressed in this work because of scope reasons.

- Are there alternative notions of explanations and repairs?
- How do repairs interact with ontological reasoning with RDFS and OWL?
- How can we address recursive dependencies of shape targets and repairs?
- Given the high complexity of the repair problem, how can we improve the scalability in practice?

– How far does the integration of data-driven methods for repair strategies improve the quality of a KG from a practical point of view?

These questions are future work outside of the scope limits of this thesis.

4 Research Methodology and Approach

The research methodology for this thesis addresses the combination of theoretical and applied research. We develop our work incrementally by starting with the formal basis on the theoretical part and then going towards implementation and experiments to apply our approach to practical use cases.

4.1 Methodology

The initial question of this thesis is how we can identify and repair quality problems in enterprise knowledge graphs. We started with a gap analysis regarding this question and identified open issues that we will address. We then determined the main research questions based on a high level understanding of this problem. The research questions are stated to clearly define and narrow down the scope in the broad field of knowledge graph quality. The main research questions of this thesis will be addressed incrementally, with each answer to a question providing the context for the next question. At the beginning we will address *RQ-1* and provide a formal basis and a prototypical implementation. Then we follow with *RQ-2* and *RQ-3*, which have a focus on applied research in the context of specific use cases, where we will verify the applicability of our approach and incorporate the insights into our work.

– For *RQ-1*, we will investigate how to calculate repairs for SHACL constraint violations, while considering practical aspects, like minimal changes done by repairs to preserve as much as possible of the existing data. We will also discuss the computational complexity of repairs and choose an appropriate technology for implementation.
– For *RQ-2*, we will consider the scenario where users want to provide preferences for different repair choices. *RQ-2* will be answered in the context of use cases, where we will specifically answer the question of what users would accept as a high quality repair in such a scenario.
– For *RQ-3*, we will look into integration of data-driven methods for inferring values in knowledge graphs as an extension of the outcomes of *RQ-2*. Answering *RQ-3* will allow us to integrate formal repairs with data-driven analysis methods into a hybrid approach and we will provide a prototypical implementation. *RQ-3* will also be answered in the context of use cases.

The proposed methodology aims to develop a system for knowledge graph repair built on a strong formal basis and developed in practical scenarios. To the best of our knowledge, our approach is novel in the way how we address data quality for knowledge graphs and our implementation is an innovation where no similar solutions currently exist.

5 Evaluation

The evaluation of our work has 3 aspects. First, we will verify the correctness of the repairs against the SHACL validation. After computing a repair, the repaired data graph must always validate against the constraints. If it is not possible to repair all the violations, the repairs provide a maximal consistent data graph and report the targets which were not repaired. This evaluation can be done by using a SHACL processor for validation and can be tested automatically. We will provide unit test cases and also test against the published SHACL test suite [1]. Second, the program needs to return repairs with only minimal changes to the original data graph from all the repairs it can compute. We aim to solve this formally on the technical level by using (existing) optimization implementations which always identify the optimal repairs. Third, the approach developed is intended to be applied in practice. We will select real-world use cases and evaluate how well our approach can be used to solve them from a qualitative point of view. An important question is to understand what users would accept as repairs for these use cases in the sense that it improves the quality of the data graph from a real-world perspective. We will then evaluate if such repairs can be formally represented using repair strategies. Currently, we plan to look into two use cases for repair strategies.

- Automatic processing of contract data, which needs data consistency to satisfy legal requirements originating in the GDPR [2]. We will use real-world contract data and introduce inconsistencies to simulate system defects.
- Modifying existing OWL ontologies to make them compliant with a given use case-specific OWL fragment, while preserving as much as possible of the original ontology in a syntactic and semantic sense. We will select existing and publicly available ontologies as data for this use case.

Finally, for evaluating the integration of data-driven methods we will select another use case, which contains an appropriate data graph to explore missing values, pick preferred choices and identify outliers. We will evaluate how we can integrate these analyses as part of the repair strategies.

6 Results

The research questions are intended to incrementally build up a knowledge graph repair system. We will publish intermediary results and provide implementations for proof of concept and practical experiments that complement the theoretical work and build up a unified analysis and repair system. Currently, we have published two intermediary results as conference papers.

- In *Reasoning about Explanations for Non-validation in SHACL* [1] we explain non-validation using the notion of a repair as a collection of additions and

[1] https://w3c.github.io/data-shapes/data-shapes-test-suite/
[2] https://gdpr.eu/what-is-gdpr/

deletions whose application on a data graph results in a repaired graph that satisfies a set of SHACL constraints. This publication answers research question *RQ1.1* and provides the foundation for the next steps.
– In *Repairing SHACL Constraint Violations using Answer Set Programming* [2] we propose an algorithm to compute repairs by encoding the repair problem into an answer set program. We introduce several optimisations to the program that aim to maximise the benefit for practical scenarios. This publication answers research question *RQ1.2*, builds on the previous outcomes and is the first step towards practical experiments.

With these two contributions, we already gained insights into the topic of this thesis and shared them with the research community.

7 Conclusions and next Steps

Our work aims to contribute to research by providing a formal foundation and a prototypical implementation to compute repairs for knowledge graphs based on SHACL constraints. The next steps will cover user preferences and the integration of data-driven methods.

– For user preferences, we intend to publish the results with one publication for each use case described in section 5, where we share insights into what kind of repairs users would accept as high-quality repairs.
– For integrating data-driven methods, we still need to identify an appropriate use case and then see how we can integrate with data-driven methods. Here we will publish the results in one publication.

With our work, we contribute to improving the quality of large data graphs, especially in an enterprise knowledge graph scenario, by providing intelligent knowledge graph repair.

Acknowledgments. This Ph.D. is done under the supervision of Axel Polleres (Ph.D. advisor, Vienna University of Economics and Business) and Shqiponja Ahmeta (Vienna University of Technology).

References

1. Ahmetaj, S., David, R., Ortiz, M., Polleres, A., Shehu, B., Simkus, M.: Reasoning about Explanations for Non-validation in SHACL. In: Bienvenu, M., Lakemeyer, G., Erdem, E. (eds.) Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021. pp. 12–21 (2021). https://doi.org/10.24963/kr.2021/2
2. Ahmetaj, S., David, R., Polleres, A., Šimkus, M.: Repairing SHACL Constraint Violations Using Answer Set Programming. In: The Semantic Web – ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings. p. 375–391. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-19433-7_22

3. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edn. (2011)
4. Baader, F., Koopmann, P., Kriegel, F., Nuradiansyah, A.: Optimal ABox Repair w.r.t. Static EL TBoxes: From Quantified ABoxes Back To ABoxes. In: The Semantic Web: 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 – June 2, 2022, Proceedings. p. 130–146. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-06981-9_8
5. Bertossi, L.: Database repairing and consistent query answering. Synthesis Lectures on Data Management **3**(5), 1–121 (2011)
6. Bogaerts, B., Jakubowski, M., Van den Bussche, J.: Shacl: A description logic in disguise. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 75–88. Springer (2022)
7. Bonifati, A., Fletcher, G., Voigt, H., Yakovets, N., Jagadish, H.V.: Querying Graphs. Morgan & Claypool Publishers (2018)
8. 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
9. 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
10. Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., Duan, Z.: Knowledge Graph Completion: A Review. IEEE Access **8**, 192435–192456 (2020)
11. Galkin, M., Auer, S., Vidal, M.E., Scerri, S.: Enterprise Knowledge Graphs: A Semantic Approach for Knowledge Management in the Next Generation of Enterprise Information Systems. In: ICEIS (2). pp. 88–98 (2017)
12. Heyvaert, P., De Meester, B., Dimou, A., Verborgh, R.: Rule-driven inconsistency resolution for knowledge graph generation rules. Semantic Web **10**(6), 1071–1086 (2019)
13. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A.: Knowledge Graphs. Synthesis Lectures on Data, Semantics, and Knowledge, Morgan & Claypool Publishers (2021). https://doi.org/10.2200/S01125ED1V01Y202109DSK022
14. Kalyanpur, A.: Debugging and repair of OWL ontologies. University of Maryland, College Park (2006)
15. Knublauch, H., Kontokostas, D.: Shapes constraint language (SHACL). Tech. rep., W3C (Jul 2017), https://www.w3.org/TR/shacl/
16. Lassila, O., Hendler, J., Berners-Lee, T.: The Semantic Web. Scientific American **284**(5), 34–43 (2001)
17. Li, Y., Lambrix, P.: Repairing EL ontologies using weakening and completing. In: European Semantic Web Conference. pp. 298–315. Springer (2023)
18. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web **8**(3), 489–508 (2017)
19. Prud'hommeaux, E., Labra Gayo, J., Solbrig, H.: Shape expressions: An RDF validation and transformation language. ACM International Conference Proceeding Series **2014** (09 2014). https://doi.org/10.1145/2660517.2660523
20. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The Graph Neural Network Model. IEEETNN **20**(1), 61–80 (2009). https://doi.org/10.1109/TNN.2008.2005605