

Shapes in Graph Data: Theory and Implementation

Shqiponja Ahmetaj^{*1}, Slawomir Staworko^{*2}, Jan Van den Bussche^{*3},
and Maxime Jakubowski^{†4}

1 TU Wien, AT. shqiponja.ahmetaj@tuwien.ac.at

2 relationalAI – Berkeley, US. slawek.staworko@relational.ai

3 Hasselt University, BE. jan.vandenbussche@uhasselt.be

4 Hasselt University, BE. maxime.jakubowski@uhasselt.be

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar “Shapes in Graph Data: Theory and Implementation” (24102). The seminar brought together active expert and junior researchers, both from academia and industry, to discuss the many open problems and research directions that arise from shapes in graph data, and, more generally, flexible and expressive schema and constraint languages for graph databases. The participants informed each other on how we perceive the research area, reported on the most recent results, discussed open problems and future directions, and in particular, four working groups were formed with promising intentions to work on new research and vision papers.

Seminar March 3–8, 2024 – <https://www.dagstuhl.de/24102>

2012 ACM Subject Classification Theory of computation → Data modeling; Information systems → Integrity checking; Theory of computation → Logic and databases; Information systems → Semantic web description languages

Keywords and phrases constraint languages, data for the semantic web, graph data, schema languages

Digital Object Identifier 10.4230/DagRep.14.3.9

1 Executive Summary

Shqiponja Ahmetaj (TU Wien, Austria, shqiponja.ahmetaj@tuwien.ac.at)

Slawomir Staworko (relationalAI – Berkeley, US, slawek.staworko@relational.ai)

Jan Van den Bussche (Hasselt University, BE, jan.vandenbussche@uhasselt.be)

License © Creative Commons BY 4.0 International license

© Shqiponja Ahmetaj, Slawomir Staworko, and Jan Van den Bussche

Research Area and Goals of the Seminar

One of the main reasons for the success of graph databases is that they do not require an elaborate database schema, with accompanying integrity constraints, to be set up in advance. In these classical applications, constraints and schemas are mainly *descriptive*, having as purpose to support the mental map from the real world to the data to be managed in the database. However, the emergence of graph databases is accompanied by a paradigm shift towards new applications where schemas and constraints are used for a *prescriptive* purpose. Here, the goal is to establish a contract between the database and its users, which provides guarantees on the structure and form of data provided. This shift has led to the development of a new class of formalisms based on the notion of *shapes*. Shapes are constraints on nodes

* Editor / Organizer

† Editorial Assistant / Collector



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

Shapes in Graph Data: Theory and Implementation, *Dagstuhl Reports*, Vol. 14, Issue 3, pp. 9–30

Editors: Shqiponja Ahmetaj, Slawomir Staworko, and Jan Van den Bussche



DAGSTUHL
REPORTS

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

in the graph that impose or forbid structural patterns (involving paths, edges, labels, and constant values). Naturally, then, a novel, prescriptive notion of schema emerges, consisting of a set of shapes, together with a targeting mechanism that specifies which nodes should satisfy which shapes. In the world of RDF graphs, two main shape-based formalisms have been proposed: *SHACL* (Shapes Constraint Language), standardized by the W3C, and *ShEx* (Shape Expression schemas). In the world of property graphs (PGs), different systems have their own data definition languages, such as Cypher or GSQL. Moreover, there are recent formal approaches to define schemas for property graphs such as PG-Schema and PG-Keys. The main aim of the Dagstuhl Seminar was to bring together active researchers, both from academia and industry, to report on the most recent results, to discuss the many open problems and research directions that arise from shapes, constraints, and schemas for graph databases, and to initiate new research.

Organization and Outcomes

The organisers created a schedule based on the entries from a Google document set up before the seminar, inviting participants to add talks, demos, and research topics. The seminar began with a round of introductions, where participants also asked questions they wanted to be answered during the seminar. The final schedule included 18 contributed talks and 6 short presentations on potential research and discussion topics.

As a major result from the seminar, four working groups were formed on the topics:

1. *What is used in practice for graph data abstractions? What is needed in practice for graph data abstractions?* The group formation was inspired by related questions posed by many participants during the opening introductory round on the first day of the seminar. Several research challenges were discussed and addressing them will call for opening new human-centered research lines in the data management community and beyond.
2. *Repairs and explanations in knowledge graph data management systems in the presence of shape constraints.* The group discussed the problem of assessing and managing data quality in knowledge graphs (KGs). This is a long-standing issue that attracts significant attention both in industry and academia. The new proposals on schemas and shape languages for KGs have introduced new challenges, which involve new methods to verify their validity, to deal with inconsistency, and repair the inconsistent data.
3. *Relating 6NF (Sixth Normal Form) and PG-Schema.* In this working group, two main questions were discussed: (1) Can we show in a systematic manner how schemas for property graphs, as expressed in the proposals of PG-Schema and PG-Keys, can be represented relationally, obtaining highly decomposed (6NF) schemas with key constraints and inclusion constraints such as foreign keys? (2) Can the intent of a graph database application be formalized in a suitable variant of EER (extended Entity-Relationship) diagrams?
4. *Convergence of graph data models and schemas.* The goal of the group was to understand the commonalities and differences between RDF and LPG (labelled property graphs), and their corresponding schema languages, ShEx and SHACL for RDF, and PG-Schema for LPG. The aim is to identify a common core (a small but useful common sublanguage, easily expressible in all three formalisms) and a common superlanguage (a language that captures all three formalisms, yet remains manageable).

The organisers regard the seminar as a very successful scientific event. Members of each working group expressed a clear commitment to staying connected to further investigate these topics. The first two groups specify a vision paper as a specific goal and the result of the group's future efforts and the second two groups aim to produce research papers.

The organisers are grateful to the Scientific Directorate and to the staff for supporting in making this seminar possible.

2 Table of Contents

Executive Summary

<i>Shqiponja Ahmetaj, Slawomir Staworko, and Jan Van den Bussche</i>	9
--	---

Overview of Talks

Explanations and Repairs for Non-Validation in SHACL <i>Shqiponja Ahmetaj</i>	13
SHACL shapes extraction <i>Anastasia Dimou</i>	14
Schema Discovery for Property Graphs <i>Stefania Dumbrova and Angela Bonifati</i>	15
SHACL and SPARQL to detect inconsistencies in Wikidata <i>Nicolas Ferranti</i>	16
PG-Keys: An Introduction <i>George Fletcher</i>	17
Scalable Extraction of Shapes from Large Knowledge Graphs <i>Katja Hose</i>	17
Data Provenance for SHACL <i>Maxime Jakubowski</i>	18
Decision Problems in SHACL <i>George Konstantinidis and Fabio Mogavero</i>	18
Introduction to ShEx <i>José Emilio Labra Gayo</i>	19
ShEx and SHACL compared <i>José Emilio Labra Gayo</i>	19
Learning Schemas from Typed Graphs <i>Aurélien Lemay</i>	20
PG-Schema: An introduction <i>Filip Murlak</i>	20
An epistemic approach to model uncertainty in RegGXPath data-graphs <i>Nina Pardal</i>	21
The different “Shapes” of RDF(S) and OWL: a fragmented history <i>Axel Polleres</i>	21
SHACL vs PG-Schema <i>Ognjen Savkovic</i>	22
Towards a SHACL Validator under the Well-founded Semantics <i>Mantas Simkus and Cem Okumus</i>	22
How the Wikidata Community Uses ShEx <i>Katherine Thornton</i>	23
Introduction to SHACL <i>Jan Van den Bussche</i>	23

Primer on GQL graph types <i>Hannes Voigt</i>	24
Working groups	
“What is used in practice for graph data abstractions? What is needed in practice for graph data abstractions?”: Working group report <i>Angela Bonifati, Anastasia Dimou, Stefania Dumbrova, George Fletcher, Katja Hose, George Konstantinidis, Aurélien Lemay, Wim Martens, Nina Pardal, Liat Peterfreund, Katherine Thornton, Maria-Esther Vidal, and Hannes Voigt</i>	24
Repairs and explanations in knowledge graph data management systems in the presence of shape constraints <i>Anastasia Dimou, Shqiponja Ahmetaj, Nicolas Ferranti, Maxime Jakubowski, José Emilio Labra Gayo, Cem Okulmus, Nina Pardal, Ognjen Savkovic, and Mantas Simkus</i>	26
Relating 6NF and PG-Schema <i>Benoit Groz, Jan Hidders, Nina Pardal, Slawomir Staworko, and Piotr Wiecek</i>	27
Convergence of graph data models and schemas <i>Filip Murlak and Jan Hidders</i>	28
Participants	30

3 Overview of Talks

3.1 Explanations and Repairs for Non-Validation in SHACL

Shqiponja Ahmetaj (TU Wien, AT)

License © Creative Commons BY 4.0 International license
© Shqiponja Ahmetaj

Joint work of Shqiponja Ahmetaj, Robert David, Magdalena Ortiz, Axel Polleres, Bojken Shehu, Mantas Šimkus
Main reference Shqiponja Ahmetaj, Robert David, Magdalena Ortiz, Axel Polleres, Bojken Shehu, Mantas Šimkus: “Reasoning about Explanations for Non-validation in SHACL”, in Proc. of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021, pp. 12–21, 2021.

URL <https://doi.org/10.24963/KR.2021/2>


The Shapes Constraint Language (SHACL) is a W3C standardized language for describing and validating constraints over RDF graphs. The SHACL specification describes the so-called validation reports, which are meant to explain to the users the outcome of validating an RDF graph against a collection of shape constraints. Specifically, explaining the reasons why the input graph does not satisfy the constraints is challenging. Inspired by works on logic-based abduction and database repairs, we study in [1] the problem of explaining non-validation of SHACL constraints. In particular, in our framework non-validation is explained using the notion of a repair, i.e., a collection of additions and deletions whose application on an input graph results in a repaired graph that does satisfy the given SHACL constraints. We define a collection of decision problems for reasoning about explanations, possibly restricting to explanations that are minimal with respect to cardinality or set inclusion. We provide a detailed characterization of the computational complexity of those reasoning tasks, including the combined and the data complexity. We then propose in [2] an algorithm to compute repairs for non-recursive SHACL, the largest fragment of SHACL that is fully defined in the specification. More precisely, we encode the explanation problem – using Answer Set Programming (ASP) – into a logic program, the answer sets of which correspond to repairs. We then study a scenario where it is not possible to simultaneously repair all the targets, which may be often the case due to overall unsatisfiability or conflicting constraints. We introduce a relaxed notion of validation, which allows to validate a (maximal) subset of the targets and adapt the ASP translation to take into account this relaxation. Our implementation in Clingo is – to the best of our knowledge – the first implementation of a repair generator for SHACL.

References

- 1 Shqiponja Ahmetaj, Robert David, Magdalena Ortiz, Axel Polleres, Bojken Shehu, and Mantas Šimkus, *Reasoning about Explanations for Non-validation in SHACL*, in *18th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 12–21, 2021, doi: 10.24963/KR.2021/2.
- 2 Shqiponja Ahmetaj, Robert David, Axel Polleres, and Mantas Šimkus, *Repairing SHACL Constraint Violations Using Answer Set Programming*, in *21st International Semantic Web Conference*, pp. 375–391, Springer, 2022, doi: 10.1007/978-3-031-19433-7_22.

3.2 SHACL shapes extraction

Anastasia Dimou (KU Leuven, BE)

License  Creative Commons BY 4.0 International license
© Anastasia Dimou

Joint work of Anastasia Dimou, Xuemin Duan, David Chaves-Fraga

Main reference Xuemin Duan, David Chaves-Fraga, Olivier Derom, Anastasia Dimou: “SCOOP All the Constraints’ Flavours for Your Knowledge Graph”, in Proc. of the The Semantic Web – 21st International Conference, ESWC 2024, Hersonissos, Crete, Greece, May 26-30, 2024, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 14665, pp. 217–234, Springer, 2024.

URL https://doi.org/10.1007/978-3-031-60635-9_13

Defining shapes for the validation of RDF graphs is a non-trivial endeavour. While in most cases the shapes are manually defined, various methods were proposed for the extraction of shapes. In this talk, we went through the methods for extracting shapes and discussed open challenges related to the integration of shapes extracted from different sources.

Shapes are typically mined from RDF graphs [1, 2, 3, 4], and thus, their effectiveness is inherently influenced by the size and complexity of the RDF graph. However, these systems often overlook the constraints imposed by individual artifacts which contributed to the construction of RDF graphs.

RDF graphs are often constructed by applying ontology terms to heterogeneous data according to a set of mapping rules. Methods were proposed to extract SHACL shapes from the data schema [6, 7], the ontology [5] or the mapping rules [8]. However, these approaches lead to limited or incomplete constraints.

Methods were also proposed that exploit all artifacts associated with the construction of RDF graphs. SCOOP extract shapes from data schemas, ontologies, and mapping rules, and integrates the shapes extracted from each artifact into a unified shapes graph. SCOOP’s implementation was configured to extract shapes from XML Schema [9] using XSD2SHACL [6], OWL Ontologies [10] using Astrea [5], and RML mapping rules [11] using RML2SHACL [8].

SCOOP was applied to real-world use cases and experimental results. So far methods that exploit all artifacts associated with the construction of RDF outperform methods that extract shapes from RDF graphs. However, the integration of shapes often leads to inconsistencies. Such inconsistencies were discussed during the talk as well as strategies to deal with them.

References

- 1 Kashif Rabbani, Matteo Lissandrini, Katja Hose (2023) *Extraction of Validating Shapes from Very Large Knowledge Graphs*, VLDB Endowment, doi: 10.14778/3579075.3579078.
- 2 Daniel Fernandez-Álvarez, Jose Emilio Labra-Gayo, and Daniel Gayo-Avello, *Automatic extraction of shapes using sheXer*, Knowledge-Based Systems, vol. 238, 2022. doi: 10.1016/j.knosys.2021.107975.
- 3 Blerina Spahiu, Andrea Maurino, and Matteo Palmonari, *Towards Improving the Quality of Knowledge Graphs with Data-driven Ontology Patterns and SHACL*, in *Studies on the Semantic Web*, vol. 36: *Emerging Topics in Semantic Technologies*, pp. 52–66, 2018, IOS Press, doi: 10.3233/978-1-61499-894-5-103.
- 4 Nandana Mihindukulasooriya, Mohammad Rifat Ahmmad Rashid, Giuseppe Rizzo, Raul Garcia-Castro, Oscar Corcho, and Marco Torchiano, *RDF Shape Induction using Knowledge Base Profiling*, in *Proceedings of the 33rd ACM/SIGAPP Symposium On Applied Computing*, 2017, doi: 10.1145/3167132.3167341.
- 5 Andrea Cimmino, Alba Fernández-Izquierdo, and Raúl García-Castro, *Astrea: Automatic Generation of SHACL Shapes from Ontologies*, in *European Semantic Web Conference*, pp. 497–513, Springer, 2020, doi: 10.1007/978-3-030-49461-2_29.

- 6 Xuemin Duan, David Chaves-Fraga, and Anastasia Dimou, *XSD2SHACL: Capturing RDF Constraints from XML Schema*, in *Proceedings of the 12th Knowledge Capture Conference 2023*, pp. 214–222, Association for Computing Machinery, 2023, doi: 10.1145/3587259.3627565.
- 7 Herminio Garcia-Gonzalez and Jose Emilio Labra-Gayo, *XMLSchema2ShEx: Converting XML validation to RDF validation*, *Semantic Web*, vol. 11, no. 2, pp. 235–253, 2020, publisher: IOS Press.
- 8 Thomas Delva, Birte De Smedt, Sitt Min Oo, Dylan Van Assche, Sven Lieber, and Anastasia Dimou, *RML2SHACL: RDF Generation Taking Shape*, in *Proceedings of the 11th on Knowledge Capture Conference*, pp. 153–160, ACM, 2021, doi: 10.1145/3460210.3493562.
- 9 David Fallside, & Priscilla Walmsley, *XML Schema Part 0: Primer Second Edition*. (W3C,2004,10), <https://www.w3.org/TR/xmlschema-0/>
- 10 Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, & Michael Smith. *OWL 2 Web Ontology Language – Structural Specification and Functional-Style Syntax (Second Edition)*. (World Wide Web Consortium (W3C),2012, <http://www.w3.org/TR/owl2-syntax/>
- 11 Ana Iglesias-Molina, Dylan Van Assche, Julian Arenas-Guerrero, Ben De Meester, Christophe Debruyne, Sam Jozashoori, Maria Poveda, Michel Frank, David Chaves-Fraga, & Anastasia Dimou. *The RML Ontology: A Community-Driven Modular Redesign After a Decade of Experience in Mapping Heterogeneous Data to RDF*. The Semantic Web – ISWC 2023. pp. 152-175, 2023.
- 12 Xuemin Duan, David Chaves-Fraga, Oliver Derom, & Anastasia Dimou. *SCOOP all the Constraints’ Flavours for your Knowledge Graph*. The Semantic Web – ESWC2024, 2024.

3.3 Schema Discovery for Property Graphs

Stefania Dumbrava (ENSIIE – Paris, FR) and Angela Bonifati (Université Claude Bernard – Lyon, FR & IUF – Paris, FR)

License © Creative Commons BY 4.0 International license
© Stefania Dumbrava and Angela Bonifati

Joint work of Angela Bonifati, Stefania Dumbrava, Emile Martinez, Nicolas Mir, Hanâ Lbath, Fatemeh Ghasemi, Malo Jaffré, Pacome Luton, Thomas Pickles

Main reference Angela Bonifati, Stefania Dumbrava, Nicolas Mir: “Hierarchical Clustering for Property Graph Schema Discovery”, in Proc. of the 25th International Conference on Extending Database Technology, EDBT 2022, Edinburgh, UK, March 29 – April 1, 2022, pp. 2:449–2:453, [OpenProceedings.org](https://openproceedings.org/2022), 2022.

URL <https://doi.org/10.48786/EDBT.2022.39>

Property graphs are becoming pervasive in various graph processing applications using interconnected data. They allow encoding multi-labeled nodes and edges, as well as their properties, represented as key/value pairs. Although property graphs are widely used in several open-source and commercial graph databases, their schema definition is not as well-understood as that of their relational counterparts. The property graph schema discovery problem consists of extracting the underlying schema concepts and types from such graph datasets. The talk provides an overview of two recent schema discovery methods for property graphs.

The first method, MRSchema [1], builds upon Cypher queries to extract the node and edge serialization of a property graph, then leverages a MapReduce type inference system to obtain subtype and supertype information for nodes, and analyzes these to compute node hierarchies. The second method, GMMSchema [2], relies on hierarchical clustering using a Gaussian Mixture Model, which accounts for both node labels and properties, unlike

MRSchema. This allows for preventing the discovery of spurious types and achieving efficient performance without accuracy loss. Moreover, the approach supports efficient incremental schema maintenance, as showcased in the corresponding DiscoPG tool [3].

DiscoPG allows users to perform schema discovery for both static and dynamic graph datasets. Suitable visualization layouts and dedicated dashboards enable the user perception of the static and dynamic inferred schema on the node clusters, as well as the differences in runtimes and clustering quality. To our knowledge, DiscoPG is the first system to tackle the property graph schema discovery problem. As such, it supports the insightful exploration of the graph schema components and their evolving behavior, while revealing the underpinnings of the clustering-based discovery process.

References

- 1 Hanâ Lbath, Angela Bonifati, Russ Harmer: Schema Inference for Property Graphs. EDBT 2021: 499-504
- 2 Angela Bonifati, Stefania Dumbra, Nicolas Mir: Hierarchical Clustering for Property Graph Schema Discovery. EDBT 2022: 2:449-2:453
- 3 Angela Bonifati, Stefania-Gabriela Dumbra, Emile Martinez, Fatemeh Ghasemi, Malo Jaffré, Pacome Luton, Thomas Pickles: DiscoPG: Property Graph Schema Discovery and Exploration. Proc. VLDB Endow. 15(12): 3654-3657 (2022)

3.4 SHACL and SPARQL to detect inconsistencies in Wikidata

Nicolas Ferranti (Wirtschaftsuniversität Wien, AT)

License  Creative Commons BY 4.0 International license
© Nicolas Ferranti

Joint work of Nicolas Ferranti, Jairo Francisco de Souza, Shqiponja Ahmetaj, Axel Polleres
Main reference Nicolas Ferranti, Jairo Francisco de Souza, Shqiponja Ahmetaj, Axel Polleres: “Formalizing and Validating Wikidata’s Property Constraints using SHACL and SPARQL”, in *Semantic Web Journal*, 2024
URL <https://www.semantic-web-journal.net/system/files/swj3611.pdf>

In this talk, I delve into the crucial role of constraints in maintaining data integrity in knowledge graphs with a specific focus on Wikidata, one of the most extensive collaboratively maintained open data knowledge graphs on the Web. The World Wide Web Consortium (W3C) recommends SHACL as the constraint language for validating Knowledge Graphs, which comes in two different levels of expressivity, SHACL-Core, as well as SHACL-SPARQL. Despite the availability of SHACL, Wikidata currently represents its property constraints through its own RDF data model, which relies on Wikidata’s specific reification mechanism. This talk discusses whether and how the semantics of Wikidata property constraints, can be formalized using SHACL-Core, SHACL-SPARQL, as well as directly as SPARQL queries.

3.5 PG-Keys: An Introduction

George Fletcher (TU Eindhoven, NL)

License © Creative Commons BY 4.0 International license
© George Fletcher

Main reference Renzo Angles, Angela Bonifati, Stefania Dumbrova, George Fletcher, Keith W. Hare, Jan Hidders, Victor E. Lee, Bei Li, Leonid Libkin, Wim Martens, Filip Murlak, Josh Perryman, Ognjen Savkovic, Michael Schmidt, Juan F. Sequeda, Slawek Staworko, Dominik Tomaszuk: “PG-Keys: Keys for Property Graphs”, in Proc. of the SIGMOD ’21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021, pp. 2423–2436, ACM, 2021.

URL <https://doi.org/10.1145/3448016.3457561>

We give an introduction to PG-Keys, which together with PG-Schemas forms a community consensus proposal for property graph schema standards. PG-Keys enable the definition of key constraints on property graphs under different modes, which are combinations of basic restrictions that require the key to be exclusive, mandatory, and singleton. Further, PG-Keys can be defined on nodes, edges, and properties since in practice these all can represent valid entities. PG-Keys was an outcome of the Linked Data Benchmark Council’s Property Graph Schema Working Group, consisting of members from industry, academia, and ISO GQL standards group, representing the past, present, and future of the science and practice of property graph data management.

3.6 Scalable Extraction of Shapes from Large Knowledge Graphs

Katja Hose (TU Wien, AT)

License © Creative Commons BY 4.0 International license
© Katja Hose

Joint work of Kashif Rabbani, Matteo Lissandrini, Katja Hose

Main reference Kashif Rabbani, Matteo Lissandrini, Katja Hose: “SHACTOR: Improving the Quality of Large-Scale Knowledge Graphs with Validating Shapes”, in Proc. of the Companion of the 2023 International Conference on Management of Data, SIGMOD/PODS 2023, Seattle, WA, USA, June 18-23, 2023, pp. 151–154, ACM, 2023.

URL <https://doi.org/10.1145/3555041.3589723>

Shapes, may they be formulated in SHACL or ShEx, have become an important instrument for validating knowledge graphs and ensuring that their content adheres to a set of well-defined constraints. Building upon such constraints, downstream applications incl. machine learning can benefit from the ensured or increased quality of knowledge graphs. Despite their usefulness in a broad range of situations, the adoption of shapes is hampered by the fact that there so far is a lack of tools that enable efficient mining of meaningful shapes. This motivated us to develop an approach that can automatically mine shapes from very large knowledge graphs [2] while providing an effective means to identify meaningful shapes based on support and confidence. As an extension, we have developed SHACTOR [1], which offers users a graphical interface and the opportunity to directly edit and update the underlying knowledge graph based on violations and inconsistencies identified after mining the shapes. While the current approach focuses on a subset of SHACL, we are working on extending the scope by enabling ShEx and a broader range of constraints.

References

- 1 Kashif Rabbani, Matteo Lissandrini, Katja Hose. *SHACTOR: Improving the Quality of Large-Scale Knowledge Graphs with Validating Shapes*. SIGMOD Conference Companion. pp. 151-154, 2023
- 2 Kashif Rabbani, Matteo Lissandrini, Katja Hose. *Extraction of Validating Shapes from very large Knowledge Graphs*. Proc. VLDB Endow. 16(5), pp. 1023–1032, 2023

3.7 Data Provenance for SHACL

Maxime Jakubowski (Hasselt University, BE)

License © Creative Commons BY 4.0 International license
© Maxime Jakubowski

Joint work of Maxime Jakubowski, Thomas Delva, Anastasia Dimou, Jan Van den Bussche

Main reference Thomas Delva, Anastasia Dimou, Maxime Jakubowski, Jan Van den Bussche: “Data Provenance for SHACL”, in Proc. of the Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28-31, 2023, pp. 285–297, OpenProceedings.org, 2023.

URL <https://doi.org/10.48786/EDBT.2023.23>

Using SHACL, we present the notion of neighborhood of a node v satisfying a given shape in a graph G . This neighborhood is a subgraph of G , and provides data provenance of v for the given shape. We establish a correctness property for the obtained provenance mechanism, by proving that neighborhoods adhere to the Sufficiency requirement articulated for provenance semantics for database queries. As an additional benefit, neighborhoods allow a novel use of shapes: the extraction of a subgraph from an RDF graph, the so-called shape fragment. We compare shape fragments with SPARQL queries. We discuss implementation strategies for computing neighborhoods, and present initial experiments demonstrating that our ideas are feasible.

References

- 1 Thomas Delva, Anastasia Dimou, Maxime Jakubowski, Jan Van den Bussche. *Data Provenance for SHACL*. Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28–31, 2023

3.8 Decision Problems in SHACL

George Konstantinidis (University of Southampton, GB) and Fabio Mogavero (University of Naples, IT)

License © Creative Commons BY 4.0 International license
© George Konstantinidis and Fabio Mogavero

Joint work of George Konstantinidis, Fabio Mogavero, Paolo Pareti

Main reference Paolo Pareti, George Konstantinidis, Fabio Mogavero: “Satisfiability and containment of recursive SHACL”, J. Web Semant., Vol. 74, p. 100721, 2022.

URL <https://doi.org/10.1016/J.WEBSEM.2022.100721>

The Shapes Constraint Language (SHACL) is a W3C recommendation language used for validating RDF data by examining specific shapes within graphs. While previous research has largely focused on validation, the standard decision problems of satisfiability and containment have only been investigated for simplified versions of SHACL. In this talk, we offer a view of SHACL’s diverse features and introduce Shape Constraint Logic (SCL), an extension of a first-order language that accurately captures SHACL’s semantics. Additionally, we present MSCL, a second-order extension of SCL, which allows us to define, in a unified formal logic framework, the main recursive semantics of SHACL. Using this framework, we provide a detailed analysis of (un)decidability and complexity for the satisfiability and containment decision problems across different SHACL fragments. Notably, while both problems are undecidable for the complete language, we identify decidable combinations of interesting features, even amidst recursion.

References

- 1 Paolo Pareti, George Konstantinidis, Fabio Mogavero, Timothy J. Norman. *SHACL Satisfiability and Containment*. ISWC (1) 2020: 474-493
- 2 Paolo Pareti, George Konstantinidis, Fabio Mogavero. *Satisfiability and Containment of Recursive SHACL*. J. Web Semant. 74: 100721 (2022)

3.9 Introduction to ShEx

José Emilio Labra Gayo (University of Oviedo, ES)

License © Creative Commons BY 4.0 International license
© José Emilio Labra Gayo

Main reference Eric Prud'hommeaux, José Emilio Labra Gayo, Harold R. Solbrig: "Shape expressions: an RDF validation and transformation language", in Proc. of the 10th International Conference on Semantic Systems, SEMANTiCS 2014, Leipzig, Germany, September 4-5, 2014, pp. 32-40, ACM, 2014.

URL <https://doi.org/10.1145/2660517.2660523>

Shape Expressions (ShEx) is a concise and human-readable language to describe and validate RDF data. In this talk, we present an introduction to the ShEx language, describing the main features of the language and how it can be used for RDF validation. We started with a short motivation about the need of ShEx and we later presented the notion of shape in ShEx as well as the evolution of the language and its main motivation and features.

References

- 1 Eric Prud'hommeaux, Jose E. Labra Gayo, Harold Solbrig, *Shape expressions: an RDF validation and transformation language*. 10th International Conference on Semantic Systems, 32-40, Leipzig, Germany
- 2 Jose E. Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas, *Validating RDF data*. Springer Nature, 2017

3.10 ShEx and SHACL compared

José Emilio Labra Gayo (University of Oviedo, ES)

License © Creative Commons BY 4.0 International license
© José Emilio Labra Gayo

Main reference José Emilio Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas: "Validating RDF Data", Morgan & Claypool Publishers, 2017.

URL <https://doi.org/10.2200/S00786ED1V01Y201707WBE016>

In this talk, we presented a comparison between ShEx (Shape Expressions) and SHACL (Shapes Constraint Language). Although they have several common features, there are several differences. An important one is the motivation for their design: while ShEx has more emphasis on Description and Validation, SHACL has more emphasis on Constraints and Validation which makes ShEx schemas more similar to a grammar that defines RDF data topologies, which SHACL shapes are more similar to a conjunction of constraints about RDF data.

References

- 1 Jose E. Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas *Validating RDF data*. Springer Nature, 2018

3.11 Learning Schemas from Typed Graphs

Aurélien Lemay (INRIA Lille, FR)

License  Creative Commons BY 4.0 International license
© Aurélien Lemay

In this talk, I present a learning algorithm for learning simple Shex expressions from typed graphs. These expressions do not include disjunction, counting, or negation. We demonstrate that while learning from a single typed graph is straightforward, the problem becomes more intricate for multi-typed graphs. This complexity arises partly due to the introduction of types. We isolate specific cases that lead to these difficulties and identify conditions under which the problem remains tractable (polynomial time) or becomes intractable.

References

- 1 Benoît Groz, Aurélien Lemay, Slawek Staworko, Piotr Wiecek: Inference of Shape Graphs for Graph Databases. ICDT 2022: 14:1–14:20

3.12 PG-Schema: An introduction

Filip Murlak (University of Warsaw, PL)

License  Creative Commons BY 4.0 International license
© Filip Murlak

Main reference Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Stefan Plantikow, Ognjen Savkovic, Michael Schmidt, Juan Sequeda, Slawek Staworko, Dominik Tomaszuk, Hannes Voigt, Domagoj Vrgoc, Mingxi Wu, Dusan Zivkovic: “PG-Schema: Schemas for Property Graphs”, CoRR, Vol. abs/2211.10962, 2022.

URL <https://doi.org/10.48550/ARXIV.2211.10962>

Property graphs have reached a high level of maturity, witnessed by multiple robust graph database systems as well as the ongoing ISO standardization effort aiming at creating a new standard Graph Query Language (GQL). Yet, despite documented demand, schema support is limited both in existing systems and in the first version of the GQL Standard. It is anticipated that the second version of the GQL Standard will include a rich DDL. Aiming to inspire the development of GQL and enhance the capabilities of graph database systems, we propose PG-Schema, a simple yet powerful formalism for specifying property graph schemas. It features PG-Types with flexible type definitions supporting multi-inheritance, as well as expressive constraints based on the recently proposed PG-Keys formalism.

3.13 An epistemic approach to model uncertainty in RegGXPath data-graphs

Nina Pardal (University of Sheffield, GB)

License © Creative Commons BY 4.0 International license
© Nina Pardal

Joint work of Sergio Abriola, Santiago Cifuentes, Maria Vanina Martinez, Nina Pardal, Edwin Pin

Main reference Sergio Abriola, Santiago Cifuentes, Maria Vanina Martinez, Nina Pardal, Edwin Pin: “An epistemic approach to model uncertainty in data-graphs”, *Int. J. Approx. Reason.*, Vol. 160, p. 108948, 2023.

URL <https://doi.org/10.1016/J.IJAR.2023.108948>

Graph databases are becoming widely successful as data models that allow to effectively represent and process complex relationships among various types of data. Data-graphs are particular types of graph databases whose representation allows both data values in the paths and in the nodes to be treated as first class citizens by the query language. As with any other type of data repository, data-graphs may suffer from errors and discrepancies with respect to the real-world data they intend to represent. In this work, we explore the notion of probabilistic unclean data-graphs, in order to capture the idea that the observed (unclean) data-graph is actually the noisy version of a clean one that correctly models the world but that we know only partially. As the factors that lead to such a state of affairs may be many, e.g., all different types of clerical errors or unintended transformations of the data, and depend heavily on the application domain, we assume an epistemic probabilistic model that describes the distribution over all possible ways in which the clean (uncertain) data-graph could have been polluted. Based on this model we define two computational problems: data cleaning and probabilistic query answering and study for both of them their corresponding complexity when considering that the polluting transformation of the data-graph can be caused by either removing (subset), adding (superset), or modifying (update) nodes and edges. For data cleaning, we explore restricted versions when the transformation only involves updating data-values on the nodes.

3.14 The different “Shapes” of RDF(S) and OWL: a fragmented history

Axel Polleres (Wirtschaftsuniversität Wien, AT)

License © Creative Commons BY 4.0 International license
© Axel Polleres

Since the introduction of the Semantic Web in the late 90s, schema and ontology languages to describe the schema of what we now call “Knowledge Graphs” have played a central role. The semantic basis of these ontology languages have – historically – been based on formalisms such as Frame Logic, Description Logics as well as Datalog. The syntactic representations of Schema axioms is integrated in Knowledge Graphs by representations of axioms in RDF, using the W3C standardised RDF, RDFS and OWL vocabularies. OWL and RDFS can therefore be both seen as logical languages, but also simply as RDF vocabularies, a constrained use of which allows us to “encode” terminological axioms as part of an RDF (knowledge) graph. Yet, an unconstrained use of these vocabularies yields obviously “unintuitive” graphs. In this short talk/paper we would like to discuss two questions, namely: (a) is there too much syntactic freedom in RDF and OWL? (b) (how) can useful syntactic fragments of OWL and RDFS usage be captured by constraints and shapes? In the course of (b) we also aim at providing an “historical” overview of (semantic and syntactic) OWL and RDFS fragments from the literature.

3.15 SHACL vs PG-Schema


Ognjen Savkovic (*Freie Universität Bozen, IT*)

License  Creative Commons BY 4.0 International license
© Ognjen Savkovic

We define SHACL abstract syntax and propose two semantics for the recursive case, and show their differences. We identify four selected issues that reflect differences between PG-Schema and SHACL. In particular, we discuss: (1) differences in labels for SHACL shape expressions and PG-Schema labels; (2) the support for negation in expressions; (3) quantification over edges and cardinality constraints; and finally (4) differences in open and closed constraints for both formalisms.

3.16 Towards a SHACL Validator under the Well-founded Semantics

Mantas Simkus (*TU Wien, AT*) and Cem Okulmus (*University of Umeå, SE*)

License  Creative Commons BY 4.0 International license
© Mantas Simkus and Cem Okulmus

W3C has recently introduced SHACL as a new standard for integrity constraints on RDF graphs. Unfortunately, the standard defines the semantics of *non-recursive* constraints only, which has spurred recent research efforts into finding a suitable, mathematically crisp semantics for constraints with cyclic dependencies. To this end, Corman et al. [4] introduced a semantics related to *supported models* known in logic programming, while Andreşel et al. [1] presented a semantics based on *stable models* known in *Answer Set Programming (ASP)*. In [2], the authors argue that recursive SHACL can be naturally equipped with a semantics inspired in the *well-founded semantics* for recursive logic programs with default negation [2]. This semantics is not only intuitive, but it is also computationally tractable, unlike the previous proposals. In this talk and demo, we review the well-founded semantics of SHACL and present an implementation of a new validator for this semantics. The implementation combines multiple technologies in order to obtain good efficiency and coverage of the features of the SHACL standard. In particular, our ShaWell¹ system uses a sophisticated strategy to issue a series of SPARQL queries over an RDF triple store, whose results are post-processed to obtain a validation outcome. For non-recursive SHACL constraints, this yields a system that is largely compliant to the SHACL standard. In case of recursion, ShaWell additionally employs a deductive database engine DLV to evaluate a logic program that is produced as part of the validation process. In this way, the SHACL validation task is reduced to the problem of evaluating an ordinary logic program under the well-founded semantics. This method is similar to the one in [3], where a SAT solver was used for handling the supported model semantics from [4].

References

- 1 Medina Andreşel, Julien Corman, Magdalena Ortiz, Juan L. Reutter, Ognjen Savkovic, and Mantas Šimkus. Stable model semantics for recursive SHACL. In *WWW '20: The Web Conference 2020*, pages 1570–1580. ACM / IW3C2, 2020.

¹ <https://github.com/cem-okulmus/shawell>

- 2 Adrian Chmurovič and Mantas Šimkus. Well-founded semantics for recursive SHACL. In Mario Alviano and Andreas Pieris, editors, *Proceedings of the 4th International Workshop on the Resurgence of Datalog in Academia and Industry (Datalog-2.0 2022)*, Genova-Nervi, Italy, September 5, 2022, volume 3203 of *CEUR Workshop Proceedings*, pages 2–13. CEUR-WS.org, 2022.
- 3 Julien Corman, Fernando Florenzano, Juan L. Reutter, and Ognjen Savkovic. Validating SHACL constraints over a SPARQL endpoint. In *Proc. of ISWC 2019*, volume 11778 of *LNCS*, pages 145–163. Springer, 2019.
- 4 Julien Corman, Juan L. Reutter, and Ognjen Savkovic. Semantics and validation of recursive SHACL. In *Proc. of ISWC 2018*, volume 11136 of *LNCS*, pages 318–336. Springer, 2018.

3.17 How the Wikidata Community Uses ShEx

Katherine Thornton (Yale University Library – New Haven, US)

License © Creative Commons BY 4.0 International license
© Katherine Thornton

The Wikidata community announced the debut of the schema namespace in 2019. Wikidata editors contribute schemas in the Shape Expressions language to the schema namespace. As of February 2024, editors have contributed more than four hundred schemas describing domains from the life sciences, to the humanities, to computing.

Wikidata editors use schemas to communicate data models and to validate entity data. Each schema page contains a link to the ShEx2 Simple Online Validator so that editors can identify which Wikidata items are currently in conformance with a schema and which items require changes in order to be brought into conformance. Each schema has a unique identifier and support for labels and descriptions in the human languages Wikidata accommodates.

Wikidata editors have written schemas leveraging many feature of ShEx including recursion and importing one schema into another.

As awareness of Wikidata’s schema namespace grows, we anticipate more editors will author schemas, more editors will develop ShEx-based tooling for the community, and that the ecosystem of schemas that anyone can reuse and edit will continue to thrive.

3.18 Introduction to SHACL

Jan Van den Bussche (Hasselt University, BE)

License © Creative Commons BY 4.0 International license
© Jan Van den Bussche

Joint work of Jan Van den Bussche, Bart Bogaerts, Maxime Jakubowski

Main reference Bart Bogaerts, Maxime Jakubowski, Jan Van den Bussche: “Expressiveness of SHACL Features and Extensions for Full Equality and Disjointness Tests”, *Log. Methods Comput. Sci.*, Vol. 20(1), 2024.

URL [https://doi.org/10.46298/LMCS-20\(1:16\)2024](https://doi.org/10.46298/LMCS-20(1:16)2024)

We give an introduction to the W3C-recommended Shapes Constraint Language, SHACL. We present the syntax based on description logics introduced by Corman et al., and extended to full core SHACL by Jakubowski. We point out that SHACL views an RDF graph as an edge-labeled graph. This is interesting, since a working group during the seminar proposed edge-labeled graphs (with types) as a “greatest common lower bound” for RDF and property graphs. We present a generalization of SHACL in terms of general inclusions among shapes.

Under the natural semantics, and excluding closedness constraints, we remark that this generalization does not add expressive power compared to real SHACL where left-hand shapes must be targets of specific kinds only. We discuss expressiveness issues, and approaches to recursion. We touch briefly upon SHACL engines and systems research on the topic, and mention applications of shapes beyond validation.

3.19 Primer on GQL graph types

Hannes Voigt (Neo4j – Leipzig, DE)

License © Creative Commons BY 4.0 International license
 © Hannes Voigt
URL <https://www.iso.org/standard/76120.html>

ISO/IEC 39075:2024 – GQL defines “a database language for modeling structured data as a graph, and for storing, querying, and modifying that data in a graph database or other graph store”. GQL was published in April 2024. Part of GQL is the concept of a graph type. A graph type as a GQL-object type describing a graph in terms of restrictions on its labels, properties, nodes, edges, and topology. The purpose of a graph type is to constrain the set of nodes and edges that can be contained in a graph. The talk gave an introduction into GQL graph types, covering the basics of the concept, DDL operations on graph types, syntax and semantics as well as the advanced topics of key label sets and structural consistency.

4 Working groups

4.1 “What is used in practice for graph data abstractions? What is needed in practice for graph data abstractions?”: Working group report

Angela Bonifati (Université Claude Bernard – Lyon, FR & IUF – Paris, FR), Anastasia Dimou (KU Leuven, BE), Stefania Dumbrova (ENSIIE – Paris, FR), George Fletcher (TU Eindhoven, NL), Katja Hose (TU Wien, AT), George Konstantinidis (University of Southampton, GB), Aurélien Lemay (INRIA Lille, FR), Wim Martens (Universität Bayreuth, DE), Nina Pardal (University of Sheffield, GB), Liat Peterfreund (The Hebrew University of Jerusalem, IL), Katherine Thornton (Yale University Library – New Haven, US), Maria-Esther Vidal (TIB – Hannover, DE), and Hannes Voigt (Neo4j – Leipzig, DE)

License © Creative Commons BY 4.0 International license
 © Angela Bonifati, Anastasia Dimou, Stefania Dumbrova, George Fletcher, Katja Hose, George Konstantinidis, Aurélien Lemay, Wim Martens, Nina Pardal, Liat Peterfreund, Katherine Thornton, Maria-Esther Vidal, and Hannes Voigt

Usability is a perennial topic in the study of data and knowledge systems. If we look at the first volume of the ACM Transactions on Database Systems, we find already McGee’s criteria for usability for data abstractions (i.e., data models, query languages, schema languages): ease of comprehension and learning; ease of information modeling; ease of data definition and programming; and, ease of formalizability and theoretical study (McGee 1976). During the opening introductory round on the first day of the seminar, many participants mentioned these criteria as they apply to graph shapes. This cross-cutting concern led to the formation

of a working group for the seminar, focusing on the questions: What is used in practice for graph data abstractions, and how can we find this out? What is needed in practice for graph data abstractions?

Graph data abstractions were recognized early as being close to “mental structures underlying human thinking” and hence beneficial for usability of data systems (Sowa 1976). In the working group, we discussed several motivational use cases around the challenge of usability of graph data abstractions. Some of these use cases were inspired from open-source healthcare data (Johnson 2023) with different usability needs. Other use cases focused on the freely available Wikidata knowledge graph on which communities of contributors have specific knowledge-intensive tasks. A third use case is provided by the analysis of open DBpedia query logs (Bonifati et al. 2020). A fourth use case is provided by a survey on how shapes are generated and adopted by the community (Rabbani et al. 2022).

We also focused on the terminology to indicate the actors of usability, distinguishing between data graph builders, analysts, and consumers (Li et al. 2024). Are they end users? Are they a broader group of people with diverse domain expertises (and beyond, people impacted by our work in society at large)? This entails the question of identifying the right usability level for each group of people depending on several factors, such as their familiarity with graph-based human-computer interfaces, or graph-based formalisms expressing high-level abstractions, such as data models, query languages and schema languages.

To advance towards human-centered graph abstractions, we also need to look into aspects of fairness and responsibility. Graphs are conceptualizations of a domain of interest and they encode human biases that potentially have beneficial and/or harmful impacts on people. We identified two main points of bias: bias in the graph conceptualization and abstraction level, e.g., the schema of a graph; and, bias in data instances. We must develop solutions that study, detect and mitigate bias on both levels. This is especially important as it is our community that is mainly responsible for the design and engineering of these abstractions, a very impactful dimension.

This opens several new research challenges. The first is to study the suitability and impact of current graph data abstractions for human benefits, at the technical, application, and societal levels. Second, and related to this, the limitations of current approaches must be studied from human-centered perspectives. Third, to adequately address these challenges, we must also, as a research community, make use of research methodologies typically deployed in other areas, such as AI fairness, (design and use) of programming languages, the Visualization and the HCI communities. We need to look in these communities for both quantitative and qualitative methodologies. For qualitative methods we should also look outside our own discipline to fields such as social sciences and cognitive psychology. Finally, we must understand how to translate the answers of these questions to the next generation of computer and data scientists. This will require us to investigate data systems education and update existing and design new curricula at all levels of education.

Addressing these challenges will call for opening new human-centered research lines in the data management community and beyond, as we experience a broader turn in the scientific community towards placing people and society more centrally in our work.

References

- 1 Angela Bonifati, Wim Martens, Thomas Timm. An analytical study of large SPARQL query logs. *VLDB J.* 29(2–3): 655–679 (2020).
- 2 Angela Bonifati, Ugo Comignani, Emmanuel Coquery, Romuald Thion. Interactive Mapping Specification with Exemplar Tuples. *SIGMOD Conference 2017*: 667–682.

- 3 A.E.W. Johnson, L. Bulgarelli, L. Shen, et al. MIMIC-IV, a freely accessible electronic health record dataset. *Sci Data* 10, 1 (2023).
- 4 Paul Juillard, Angela Bonifati, Andrea Mauri. Interactive Graph Repairs for Neighborhood Constraints. *EDBT* 2024.
- 5 Harry X. Li, Gabriel Appleby, Camelia Daniela Brumar, Remco Chang, Ashley Suh. Knowledge Graphs in Practice: Characterizing their Users, Challenges, and Visualization Opportunities. *IEEE Trans. Vis. Comput. Graph.* 30(1): 584-594 (2024).
- 6 Matteo Lissandrini, Davide Mottin, Katja Hose, Torben Bach Pedersen. Knowledge Graph Exploration Systems: are we lost? *CIDR* 2022.
- 7 William C. McGee. On user criteria for data model evaluation. *ACM Trans. Database Syst.* 1(4): 370-387 (1976).
- 8 Kashif Rabbani, Matteo Lissandrini, Katja Hose. SHACL and ShEx in the Wild: A Community Survey on Validating Shapes Generation and Adoption. *WWW (Companion Volume)* 2022: 260-263.
- 9 John F. Sowa: Conceptual Graphs for a Data Base Interface. *IBM J. Res. Dev.* 20(4): 336-357 (1976).

4.2 Repairs and explanations in knowledge graph data management systems in the presence of shape constraints

Anastasia Dimou (KU Leuven, BE), Shqiponja Ahmetaj (TU Wien, AT), Nicolas Ferranti (Wirtschaftsuniversität Wien, AT), Maxime Jakubowski (Hasselt University, BE), José Emilio Labra Gayo (University of Oviedo, ES), Cem Okulmus (University of Umeå, SE), Nina Pardal (University of Sheffield, GB), Ognjen Savkovic (Freie Universität Bozen, IT), and Mantas Simkus (TU Wien, AT)

License © Creative Commons BY 4.0 International license

© Anastasia Dimou, Shqiponja Ahmetaj, Nicolas Ferranti, Maxime Jakubowski, José Emilio Labra Gayo, Cem Okulmus, Nina Pardal, Ognjen Savkovic, and Mantas Simkus

Context. The problem of assessing and managing data quality in knowledge graphs (KGs) is a long-standing issue that attracts significant attention both in industry and academia. The new proposals on schemas and constraints languages (such as SHACL, ShEx, PG-schema) for knowledge graphs (KGs), so-called shapes, have introduced new challenges. Since these new languages allow users to easily express complex properties over KGs, this requires new methods to verify them, deal with inconsistency, and repair the inconsistent data.

Setting. KGs are created throughout different processes and to identify the root causes of violations, we consider the so-called knowledge-based data management (KBDM) setting, which describes the creation and integration of data into the KG, a possible ontology of the KG, schema shapes, and possible relevant queries. The implementation of these systems in real-world applications yields a non-trivial situation, as several components need to be considered: the original data, the ontology, the mapping rules, the data graph, and the shape constraints. All of these components can contribute to violations in the final data graph and as such are potential candidates to be repaired. This opens several new research challenges, which need to be addressed separately.

Approach. Traditional approaches to data quality usually focus on the proposals that describe how to fix the data, i.e., which facts are missing and which facts should be deleted. That would be a viable approach in many settings where data is inserted manually and where

the information about sources is not available. On the other hand, fixing the data may not often be sufficient and one may need to look at the sources of violation that may be rooted in inaccurate source data or poor design of the mapping rules or shape constraints. Finally, one may consider scenarios where one is given queries of interests, and while data may not be of sufficient fitness overall it may be sufficient to answer the given queries.

Questions. The group discussed the possible challenges that arise when trying to obtain high-quality KGs: How do we achieve high-quality KGs? How can we obtain more tailored constraints? How to describe and repair the violations obtained through the reasoning process? How to manufacture meaningful explanations that allow us to explain what was violated and how the repair was addressed? When is it meaningful to propose fixes on data, and when on mappings, shapes, or even queries? How can we introduce a probabilistic model into the KDBM, either on data or schemas, that may provide better ways of ranking violations and repairs?

Plans. As a result of the exchange of views and taking into consideration the different backgrounds of all the participants, the group has decided to pursue a comprehensive discussion and study of the state-of-the-art inconsistency management tasks for graph data, providing as well use-cases that may shed light on the motivation behind different research questions that remain open and which may be overlooked in academic environments. We will stay in touch for further collaboration, having a vision paper as a specific horizon and the result of the group's future efforts.

4.3 Relating 6NF and PG-Schema

Benoit Groz (University Paris-Saclay – Orsay, FR), Jan Hidders (Birkbeck, University of London, GB), Nina Pardal (University of Sheffield, GB), Slawomir Staworko (relationalAI – Berkeley, US), and Piotr Wiecezorek (University of Wrocław, PL)

License © Creative Commons BY 4.0 International license
© Benoit Groz, Jan Hidders, Nina Pardal, Slawomir Staworko, and Piotr Wiecezorek

Experience has shown us that relational database schemas are very versatile and can model a variety of data modeling approaches, including property graphs. Moreover, novel techniques in query processing, such as worst-case optimal joins, or Datalog query processing, indicate that a relational approach to graph database applications is viable.

In this working group, we discussed two questions:

1. Can we show in a systematic manner how schemas for property graphs, as expressed in the proposals of PG-Schema and PG-Keys, can be represented relationally, obtaining highly decomposed (6NF) schemas with key constraints and inclusion constraints such as foreign keys?
2. Can the intent of a graph database application be formalized in a suitable variant of EER (extended Entity-Relationship) diagrams?

The established semantics of EER diagrams is through a mapping to relational schemas with constraints. The group discussed various use cases of EER diagrams with additional constructs inspired by PG-Schema, such as multivalued or optional properties. By observing the relational schemas that are obtained for these use cases, one may form a rough picture of the classes of relational constraints needed to support various graph database applications. While we believe that primary and foreign keys are two classes of constraints that can be


enforced very efficiently, we have carefully identified features of EER models that require more expressive constraints, which leaves as an important open question if they can also be efficiently enforced.

The group has found that there is a lack of consensus on the interpretation of EER diagrams, with multiple semantics proposed that differ on finer points that in the context of graph databases can have important ramifications. For instance, do relationships in EER diagrams allow for multiple links between the same pair of entity instances (multi-graphs)? Can we assume that all entities have object identifiers, thus assuming that all entities are subclasses of a single top superclass?

We pointed out that a good understanding of mappings from property graphs to EER, and from EER to relational, may yield as an added bonus, a method to visualize a class of relational database schemas as PG-Schema with PG-Keys, as well as a way to visualize PG-Schemas in the more familiar notation of EER diagrams. Members of the working group will continue to stay in touch with each other to further investigate this line of research.

4.4 Convergence of graph data models and schemas

Filip Murlak (University of Warsaw, PL) and Jan Hidders (Birkbeck, University of London, GB)

License  Creative Commons BY 4.0 International license
© Filip Murlak and Jan Hidders

Joint work of Shqiponja Ahmetaj, Iovka Boneva, Jan Hidders, Maxime Jakubowski, Jose Emilio Labra Gayo, Leonid Libkin, Wim Martens, Fabio Mogavero, Filip Murlak, Cem Okulmus, Axel Polleres, Ognjen Savković, Mantas Šimkus

4.4.1 Objective

The goal is to understand the commonalities and differences between RDF and LPG (labelled property graphs), and their corresponding schema languages, ShEx and SHACL for RDF, and PG-Schema for LPG. We aim to identify a *common core* (a small but useful common sublanguage, easily expressible in all three formalisms) and a *common superlanguage* (a language that captures all three formalisms, yet remains manageable).

4.4.2 How do we compare schema languages?

Schema languages can be used for different purposes: prescriptive (e.g., refuse certain updates, refuse a certain graph as input), descriptive (e.g., defining a vocabulary), deriving information, or defining patterns (types or shapes) to be used as part of a query language. From this, we have abstracted two concrete tasks: defining sets of valid graphs, and defining sets of nodes in a graph.

4.4.3 How to compare schema languages for different data models?

RDF and LPG are similar enough to make the comparison of their schema languages viable, but different enough to make direct comparison impossible. There are several workarounds. The simplest approach is to define restrictions on RDF and LPG that result in isomorphic data models. A more refined approach is to consider a canonical encoding of RDF in LPG, and vice versa. Then, the question is which constraints/patterns in the source schema language can be expressed in the target schema language over the encoded instances, and

which constraints/patterns over encodings can be translated back to the source schema language. Finally, one could consider a whole class of (relatively simple) encodings of one data model in the other.

4.4.4 Plans

A group of participants coming from all three communities has declared interest in pursuing these goals together, aiming to produce a research paper within a half-year horizon. We plan to identify a common restriction of LPG and RDF, give the semantics of the three formalisms over the restricted data model, identify a common core and evaluate it against known usecases, and design a manageable common superlanguage.

Participants

- Shqiponja Ahmetaj
TU Wien, AT
- Iovka Boneva
Université de Lille I, FR
- Angela Bonifati
Université Claude Bernard –
Lyon, FR & IUF – Paris, FR
- Anastasia Dimou
KU Leuven, BE
- Stefania Dumbrava
ENSIIE – Paris, FR
- Nicolas Ferranti
Wirtschaftsuniversität Wien, AT
- George Fletcher
TU Eindhoven, NL
- Benoit Groz
University Paris-Saclay –
Orsay, FR
- Jan Hidders
Birkbeck, University of
London, GB
- Katja Hose
TU Wien, AT
- Maxime Jakubowski
Hasselt University, BE
- George Konstantinidis
University of Southampton, GB
- José Emilio Labra Gayo
University of Oviedo, ES
- Aurélien Lemay
INRIA Lille, FR
- Leonid Libkin
University of Edinburgh, GB
- Wim Martens
Universität Bayreuth, DE
- Fabio Mogavero
University of Naples, IT
- Filip Murlak
University of Warsaw, PL
- Cem Okulmus
University of Umeå, SE
- Nina Pardal
University of Sheffield, GB
- Liat Peterfreund
The Hebrew University of
Jerusalem, IL
- Axel Polleres
Wirtschaftsuniversität Wien, AT
- Ognjen Savkovic
Freie Universität Bozen, IT
- Mantas Simkus
TU Wien, AT
- Slawomir Staworko
relationalAI – Berkeley, US
- Katherine Thornton
Yale University Library –
New Haven, US
- Jan Van den Bussche
Hasselt University, BE
- Maria-Esther Vidal
TIB – Hannover, DE
- Hannes Voigt
Neo4j – Leipzig, DE
- Piotr Wiecek
University of Wrocław, PL

