

SHACL Validation and Static Analysis in Presence of Ontologies: an Overview

Anouk Michelle Oudshoorn¹

¹*Institute of Logic and Computation, TU Wien*

Abstract

In this work, we provide an overview of our recent work on SHACL validation, satisfiability and containment [1, 2, 3], and specifically relate it to the setting in which SHACL constraints are paired with an ontology. This allows for combining employing database constraints to detect inconsistencies with domain knowledge to reduce gaps in the data: both standard techniques to address current data plagued by quality issues.

Keywords

SHACL, OWL, Validation, Satisfiability, Containment

1. Introduction

The Shape Constraint Language (SHACL) [4] and Web Ontology Language (OWL) [5] are two prominent W3C standards for managing RDF data, a graph-based data model of the Web [6]. These standards are based on fundamentally different assumptions and designed to be complementary. OWL was standardised shortly after RDF, with the key aim of enhancing RDF datasets with domain knowledge that enables the inference of missing facts from potentially incomplete data graphs: OWL enables to capture repeating or generalisable knowledge in a precise manner. This is because OWL and its profiles are based on *Description Logics (DLs)* [7].

Like other classical logics, DLs or OWL make the *open-world assumption (OWA)*. Intuitively means that the data only presents an incomplete description of the domain of interest: it asserts facts that are known to be true, but does not rule out that additional facts may also be true, as long as they are consistent with the current world description. OWL has been adopted in a wide range of applications over the years, and thousands of OWL ontologies have been developed (see e.g., [8]). SHACL, in contrast, was created for a different purpose: to describe and validate constraints on datasets. The main task of interest is *validation* of a given a set of constraints paired with a selection of *target* nodes or concepts from a given graph. Unlike OWL, SHACL operates under the *closed-world assumption (CWA)*: it assumes that the given data graph is complete, and validators evaluate the constraints over the input graph as is.

In this extended abstract, we aim to give an overview of our most recent work on SHACL [1, 2, 3], with a focus on the combination with ontologies: what can be considered a natural semantics for validation of SHACL in presence of OWL ontologies? For which fragments are concrete techniques known? And what do we know about static analysis problems like containment and satisfiability for SHACL, and specifically in presence of ontologies? In this way, we set the stage for a broader adaptation of SHACL to detect data-inconsistencies, while at the same time filtering out the more straightforward gaps, already taken care of because of the consideration of domain knowledge.

2. Validation

The first natural question we address is how to do validation in the presence of both OWL ontologies and SHACL constraints. That is, if we have a possibly incomplete graph and ontological knowl-

Logical approaches to handling INconsistent DATA (LINDA) 2026

✉ anouk.oudshoorn@tuwien.ac.at (A. M. Oudshoorn)

🌐 <https://noukoudshoorn.github.io> (A. M. Oudshoorn)

🆔 0009-0006-4638-5948 (A. M. Oudshoorn)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

edge that implies additional facts, can we validate given SHACL constraints over graphs containing the implied facts? Consider as an example a toy database of pet owners containing the facts $hasPetBird(linda, blu)$, $hasPet(john, ace)$; the simple constraint $petOwnerShape \leftarrow \exists hasPet.\top$, which says that everyone that has a pet is a pet owner; and the target pet owners $linda$ and $john$. Clearly, we would like to leverage the knowledge that *all pet birds are pets*, written as $hasPetBird \sqsubseteq hasPet$ in description logics, which allows us to validate both targets. This type of validation is very natural, even more so in the light of the large amount of ontologies that are already being used for describing data on the web. It is in fact envisioned in the W3C SHACL specification, which calls for graph validation in the presence of OWL entailment [4, Section 1.5], but unfortunately, does not provide guidance on how to realise this. Nevertheless, there are approaches of reconciling the two [9, 10], and concrete ontology and constraint pairs being constructed [11].

Choice of Semantics. The first major challenge we must face is that the semantics of SHACL constraints in the presence of ontologies is not obvious, as we must simultaneously account for the open-world semantics of description logics and for the closed-world view of SHACL. The knowledge in the ontology implies additional facts, which can be added to the data graph so that it satisfies all the ontological axioms. All such possible completions are *models* that must be taken into account according to the traditional OWL/description logics semantics. However, SHACL allows for negation, which makes this *certain answer semantics* too weak, and quickly results in non validation. Update for example the toy database of pet owners we considered before to $hasPet(john, ace)$, $Hamster(ace)$, but with the constraint $petOwnerShape \leftarrow \exists hasPet.\top \wedge \forall hasPet.\neg Dangerous$. Like before, this expresses that pet owners own a pet. Moreover it is required that all pets are not dangerous. For $john$ to validate the $petOwnerShape$, it is not enough to make explicit in the ontology that hamsters are not dangerous animals; it should also be enforced that any possible pet $john$ might have cannot be dangerous.

For lightweight DLs, fragments of classical Horn logics that cannot express disjunctive information, *universal models* can be obtained using standard, database-inspired *chase procedures*. These models can be used for evaluating conjunctive, navigational and graph queries in the presence of ontologies, see [12, 13, 14, 15, 16, 17] and their references. One of these options, using *minimal models* of the Skolemisation of the DL ontology, has been advocated for in the case of integrity constraints [18, 10]. But even such models give very weak semantics in formalisms with negation such as SHACL. Let us for example consider again another version of the toy database containing pet owners: assume it contains the facts $hasPet(john, ace)$, $PetOwner(john)$, $Hamster(ace)$. It is conceivable to find an axiom like $PetOwner \sqsubseteq \exists hasPet.\top$ in the accompanying ontology. Now let the *onlyHamsterShape* be given by the constraint enforcing all pets to be hamsters, $onlyHamsterShape \leftarrow \forall hasPet.Hamster$, which we want to validate for $john$. This is clearly the case for the original database. It is also true that the given database is already satisfying all given ontology axioms, so it seems there is no reason to change the validation result. However, the minimal model under the Skolem semantics adds a fresh node as a *hasPet*-child of $john$, without the label *Hamster*, changing the validation result to negative.

Austere Canonical Model. To obtain stronger and more intuitive semantics, and to avoid the problems presented in the previous example, we advocated in [19] for an *austere* canonical model in which axioms are satisfied minimally, introducing as few successors as possible without losing universality. We showed that for ontologies in DL-Lite_R—the logic underlying OWL 2 QL [20]—such a model can be represented by a so-called *immediate successor function* that describes the minimal set of facts that need to be added to satisfy the axioms at a given point of the model construction. The model itself can then be obtained in a deterministic, step-by-step fashion. This construction was first extended to the logic \mathcal{ELHI} [21], then we extended this significantly further to the more expressive Horn- \mathcal{ALCHIQ} [1], one of the largest fragments of OWL that is still contained in Horn logic. Crucially, we show that the resulting model is a *core* in the traditional database sense. This provides strong evidence in favour of our chosen semantics, since cores are often advocated as the adequate choice for languages that are not closed under homomorphisms, but satisfy the weaker property of being closed under

isomorphisms [22, 23, 24]. In the light of this relationship, our austere model construction provides a novel technique for building core models without the expensive core-checking step of traditional core chase procedures. As we point out, the same applies to some previous model constructions from the DL literature [23, 15].

Rewriting. With our semantics based on austere (i.e., core) models in place, we can tackle the problem of devising an algorithm for validation. Constructing the austere model may be infeasible in practice, since it is infinite in general. Instead, we use our finite representation of the model: as we are considering a minimised form of a canonical model, and only have finitely many axioms, the austere canonical model, if infinite, resorts to repeating patterns in the anonymous parts. This is captured by the good successor configuration function, that in this way exactly provides the puzzle pieces of which the austere canonical model is built.

Ideally, we would like to realise validation via *rewriting* (see our works [19, 21, 1] for the details). That is, we want to compile a given ontology and a set of SHACL constraints into a new set of SHACL constraints that incorporate the relevant knowledge of the ontology in such a way that the implicit facts are taken into account in validation, without having to explicitly add them to the graph. Rewriting techniques are very desirable as they open the way to reuse standard SHACL validators to perform validation in the presence of ontologies. For the rewriting, we considered stratified (recursive) SHACL under the least fixed point semantics, which coincides with the well-founded [25] and stable model semantics [26]. These semantics are all asking for some form of justification for derived shape atoms, whereas the supported model semantics asks whether it is possible to find any shape assignment. The latter has the disadvantage that in general multiple assignments exist (as might happen for the stable model semantics for non-stratified constraints), which makes it unclear how a rewriting for the supported model semantics would look like.

So, for the least fixed point semantics, we use the good successor configuration, in its role of building blocks of the austere canonical model, to derive which so-called *2-types* can exist in the anonymous part. These types intuitively represent an abstract copy of a possible object and its neighbours in the austere model of a given data graph—enriched with information about the shapes that are (not) satisfied in implied substructures. This structure is used to induce a modified set of SHACL constraints that validate over a given data graph exactly when the original constraints validate over the austere canonical model of the input graph and the ontology.

3. Static Analysis

The satisfiability problem is of major importance in the design and validation of SHACL-based solutions: as SHACL becomes more popular, substantive efforts are put into the adoption of SHACL. As part of this, we witness mining SHACL specifications from data [27, 28, 29], but how to assess the quality of these machine-generated constraints? And how to combine multiple, possibly generated, specifications? We note that the basic necessary condition here is compatibility, which boils down to satisfiability.

Given the importance of the problem, there are remarkably few results concerning its decidability and complexity. Indeed, the most notable work in this direction, [30], is very coarse. It builds on a tailored fragment of predicate logic to identify decidability and complexity bounds, but the basic logic it considers is already close to the boundary of what could potentially be decidable in the presence of cardinality constraints. Put differently, the base logic is already rather expressive (for instance already containing nominals and inverses), which means that barely any features can be added for the logic to remain decidable. That is, the positive results are mostly limited to formalisms that do not support counting, and more often than not consider unrestricted (that is, potentially infinite) graphs, even though finite graphs are a more relevant setting here.

Satisfiability. In our recent work [2], we revisit (recursive) SHACL satisfiability under the supported model semantics, which is the least restrictive semantics for SHACL: recall it just asks for the existence

of a shape assignment. That is, if a shapes graph is unsatisfiable under this semantics, it is under any semantics.

We build on Description Logics (DLs), a well-known family of languages for Knowledge Representation and Reasoning that offers decades of research in the fine-grained study of logical fragments and the effect that the interaction between different shapes of subformulas has on the complexity of reasoning. The close relationship between DLs and SHACL is well-known, and in that paper, we leverage it to paint a much finer boundary of SHACL fragments that have decidable satisfiability problems, both over unrestricted graphs and over graphs with a finite domain. Moreover, we take a closer look at the joint satisfiability of SHACL and OWL: can we find a (finite) model \mathcal{I} that validates a given shapes graph and is also a model of the given ontology? It turns out that also here, we can exploit the close connection between SHACL and DLs: joint satisfiability is as hard as deciding (finite) satisfiability in the least-expressive description logic capturing the expressivity of both the translated SHACL fragment, following the translation as presented in this work, and the DL fragment [2].

Containment or Implication. A natural next step in assessing quality of data is tackling containment or implication, for which satisfiability is a prerequisite. The main problem studied here is whether a (set of) constraints, implies or subsumes another (set of) constraints. Knowing this, or having tools to decide this, is particularly useful when trying to clean up sets of constraints: it allows to get rid of superfluous parts and thus simplify constraint sets without affecting validation outcomes.

There are multiple types of SHACL containment one can consider: from shape containment to document implication. The first problem, which addresses whether one constraint subsumes another without considering any other constraints, is studied by Leinberger et al. [31], who also base their techniques on a translation into DLs. However, as pointed out in [32], there are some issues with their translation.

In our recent work [3], we address deciding shape containment with respect to a context of constraints, and also the second type of containment: given two (possibly recursive) SHACL documents $(\mathcal{C}_1, \mathcal{T}_1)$ and $(\mathcal{C}_2, \mathcal{T}_2)$, decide whether every (finite) graph that satisfies $(\mathcal{C}_1, \mathcal{T}_1)$ also satisfies $(\mathcal{C}_2, \mathcal{T}_2)$. It turns out that for the supported model semantics, the semantics for which we could reduce SHACL satisfiability to DL satisfiability, containment is a different piece of cake: already for a base fragment of SHACL, comparable to the description logic $\mathcal{ALC}\mathcal{IO}$, we find document implication to be undecidable. For the stable model semantics this problem remains undecidable. However, decidability is regained for the well-founded semantics. This is shown via a direct translation of SHACL documents into the hybrid μ -calculus, making $\mathcal{ALC}\mathcal{IO}$ SHACL satisfiability and containment EXPTIME -complete [3].

Given these results, it is immediate it will remain undecidable to consider SHACL document implication in presence of ontologies for both the supported and the stable model semantics. However, in light of the positive results and new techniques provided for the well-founded semantics, it is conceivable that a combined translation of SHACL and DLs into hybrid μ -calculus will provide the sought after decidability results, probably making joint implication of SHACL documents under DL ontologies EXPTIME -complete for certain fragments. The concrete translations and details are left for future work.

Finally, both satisfiability and containment or implication, as static analysis tools, are prerequisites for more advanced services like optimisation, incremental validation and modularity. Also these problems, both with and without ontologies present, remain open for future work.

Declaration on Generative AI

During the preparation of this work, no generative AI was used.

References

- [1] A. M. Oudshoorn, M. Ortiz, M. Šimkus, SHACL validation in the presence of ontologies: Semantics and rewriting techniques, *Artificial Intelligence* 352 (2026) 104483. URL: <https://www>.

sciencedirect.com/science/article/pii/S0004370226000093. doi:<https://doi.org/10.1016/j.artint.2026.104483>.

- [2] A. M. Oudshoorn, SHACL satisfiability: What can we learn from DLs?, in: Proceedings of the 38th International Workshop on Description Logics (DL 2025), Opole, Poland, September 3-6, 2025, volume 4091 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025.
- [3] A. M. Oudshoorn, M. Ortiz, M. Šimkus, Static analysis of recursive SHACL, in: KR, 2026.
- [4] W3C, Shape constraint language (SHACL), Technical Report. (2017). <https://www.w3.org/TR/shacl>.
- [5] W3C, Owl 2 web ontology language, Technical Report. (2012). URL: <https://www.w3.org/TR/owl2-overview>.
- [6] W3C, Resource description framework, Technical Report. (2014). URL: <https://www.w3.org/TR/rdf11-concepts>.
- [7] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017. URL: <http://www.cambridge.org/de/academic/subjects/computer-science/knowledge-management-databases-and-data-mining/introduction-description-logic?format=PB#17zVGeWD2TZUeu6s.97>.
- [8] HL7, FHIR specification (v5.0.0) (2011). <http://www.hl7.org/fhir/>.
- [9] J. Ke, Z. G. Zacouris, M. Acosta, Efficient validation of SHACL shapes with reasoning, Proc. VLDB Endow. 17 (2024) 3589–3601. URL: <https://www.vldb.org/pvldb/vol17/p3589-acosta.pdf>. doi:10.14778/3681954.3682023.
- [10] O. Savkovic, E. Kharlamov, S. Lamparter, Validation of SHACL constraints over kgs with OWL 2 QL ontologies via rewriting, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. G. Gray, V. López, A. Haller, K. Hammar (Eds.), The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings, volume 11503 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 314–329. URL: https://doi.org/10.1007/978-3-030-21348-0_21. doi:10.1007/978-3-030-21348-0_21.
- [11] P. Gorczyca, D. Arndt, M. Diller, P. Kettmann, S. Mennicke, H. Strass, A farewell to harms: Risk management for medical devices via the riskman ontology & shapes, CoRR abs/2405.09875 (2024). URL: <https://doi.org/10.48550/arXiv.2405.09875>. doi:10.48550/ARXIV.2405.09875. arXiv:2405.09875.
- [12] M. Bienvenu, M. Ortiz, Ontology-mediated query answering with data-tractable description logics, in: W. Faber, A. Paschke (Eds.), Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures, volume 9203 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 218–307. URL: https://doi.org/10.1007/978-3-319-21768-0_9. doi:10.1007/978-3-319-21768-0_9.
- [13] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, G. Xiao, Ontop: Answering SPARQL queries over relational databases, Semantic Web 8 (2017) 471–487. URL: <https://doi.org/10.3233/SW-160217>. doi:10.3233/SW-160217.
- [14] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, D. F. Savo, The MASTRO system for ontology-based data access, Semantic Web 2 (2011) 43–53. URL: <https://doi.org/10.3233/SW-2011-0029>. doi:10.3233/SW-2011-0029.
- [15] T. Eiter, M. Ortiz, M. Šimkus, T. Tran, G. Xiao, Query rewriting for horn-shiq plus rules, in: J. Hoffmann, B. Selman (Eds.), Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada, AAAI Press, 2012, pp. 726–733. URL: <https://doi.org/10.1609/aaai.v26i1.8219>. doi:10.1609/AAAI.V26I1.8219.
- [16] M. Ortiz, S. Rudolph, M. Šimkus, Query answering in the horn fragments of the description logics SHOIQ and SROIQ, in: T. Walsh (Ed.), IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011, IJCAI/AAAI, 2011, pp. 1039–1044. URL: <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-178>. doi:10.5591/978-1-57735-516-8/IJCAI11-178.
- [17] H. Pérez-Urbina, B. Motik, I. Horrocks, Tractable query answering and rewriting under description logic constraints, J. Appl. Log. 8 (2010) 186–209. URL: <https://doi.org/10.1016/j.jal.2009.09.004>. doi:10.1016/J.JAL.2009.09.004.

- [18] B. Motik, I. Horrocks, U. Sattler, Bridging the gap between OWL and relational databases, in: Proceedings of WWW, ACM, 2007. URL: <https://doi.org/10.1145/1242572.1242681>. doi:10.1145/1242572.1242681.
- [19] S. Ahmetaj, M. Ortiz, A. M. Oudshoorn, M. Šimkus, Reconciling SHACL and ontologies: Semantics and validation via rewriting, in: K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, R. Radulescu (Eds.), ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland, volume 372 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2023, pp. 27–35. URL: <https://doi.org/10.3233/FAIA230250>. doi:10.3233/FAIA230250.
- [20] M. Krötzsch, OWL 2 profiles: An introduction to lightweight ontology languages, in: T. Eiter, T. Krennwallner (Eds.), Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings, volume 7487 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 112–183. URL: https://doi.org/10.1007/978-3-642-33158-9_4. doi:10.1007/978-3-642-33158-9_4.
- [21] A. M. Oudshoorn, M. Ortiz, M. Šimkus, Reasoning with the core chase: the case of SHACL validation over ELHI knowledge bases, in: L. Giordano, J. C. Jung, A. Ozaki (Eds.), Proceedings of the 37th International Workshop on Description Logics (DL 2024), Bergen, Norway, June 18-21, 2024, volume 3739 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: <https://ceur-ws.org/Vol-3739/paper-7.pdf>.
- [22] M. Arenas, G. Gottlob, A. Pieris, Expressive languages for querying the semantic web, *ACM Trans. Database Syst.* 43 (2018) 13:1–13:45. URL: <https://doi.org/10.1145/3238304>. doi:10.1145/3238304.
- [23] S. Borgwardt, W. Forkel, Closed-world semantics for conjunctive queries with negation over ELH-bottom ontologies, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 6131–6135. URL: <https://doi.org/10.24963/ijcai.2019/849>. doi:10.24963/ijcai.2019/849.
- [24] S. Ellmauthaler, M. Krötzsch, S. Mennicke, Answering queries with negation over existential rules, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 5626–5633. URL: <https://doi.org/10.1609/aaai.v36i5.20503>. doi:10.1609/AAAI.V36I5.20503.
- [25] C. Okulmus, M. Šimkus, SHACL validation under the well-founded semantics, in: P. Marquis, M. Ortiz, M. Pagnucco (Eds.), Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024, 2024. URL: <https://doi.org/10.24963/kr.2024/52>. doi:10.24963/KR.2024/52.
- [26] M. Andresel, J. Corman, M. Ortiz, J. L. Reutter, O. Savkovic, M. Šimkus, Stable model semantics for recursive SHACL, in: Y. Huang, I. King, T. Liu, M. van Steen (Eds.), WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, ACM / IW3C2, 2020, pp. 1570–1580. URL: <https://doi.org/10.1145/3366423.3380229>. doi:10.1145/3366423.3380229.
- [27] P. G. Omran, K. Taylor, S. J. R. Méndez, A. Haller, Learning SHACL shapes from knowledge graphs, *Semantic Web* 14 (2023) 101–121. URL: <https://doi.org/10.3233/SW-223063>. doi:10.3233/SW-223063.
- [28] D. Fernández-Álvarez, J. E. L. Gayo, D. Gayo-Avello, Automatic extraction of shapes using shexer, *Knowl. Based Syst.* 238 (2022) 107975. URL: <https://doi.org/10.1016/j.knosys.2021.107975>. doi:10.1016/J.KNOSYS.2021.107975.
- [29] K. Rabbani, M. Lissandrini, K. Hose, Extraction of validating shapes from very large knowledge graphs, *Proc. VLDB Endow.* 16 (2023) 1023–1032. URL: <https://www.vldb.org/pvldb/vol16/p1023-rabbani.pdf>. doi:10.14778/3579075.3579078.
- [30] P. Paretì, G. Konstantinidis, F. Mogavero, Satisfiability and containment of recursive SHACL, *J. Web Semant.* 74 (2022) 100721. URL: <https://doi.org/10.1016/j.websem.2022.100721>. doi:10.1016/J.WEBSEM.2022.100721.
- [31] M. Leinberger, P. Seifer, T. Rienstra, R. Lämmel, S. Staab, Deciding SHACL shape containment through description logics reasoning, in: J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu,

- A. Polleres, O. Seneviratne, L. Kagal (Eds.), The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I, volume 12506 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 366–383. URL: https://doi.org/10.1007/978-3-030-62419-4_21. doi:10.1007/978-3-030-62419-4_21.
- [32] B. Bogaerts, M. Jakubowski, J. Van den Bussche, SHACL: A description logic in disguise, in: G. Gottlob, D. Incezan, M. Maratea (Eds.), Logic Programming and Nonmonotonic Reasoning - 16th International Conference, LPNMR 2022, Genova, Italy, September 5-9, 2022, Proceedings, volume 13416 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 75–88. URL: https://doi.org/10.1007/978-3-031-15707-3_7. doi:10.1007/978-3-031-15707-3_7.