Logic-based Semantics and Query Entailment for RDFS Knowledge Graphs (Extended Abstract)

Roberto Maria Delfino, Maurizio Lenzerini and Antonella Poggi

Sapienza University of Rome, via Ariosto 25, Rome, Italy

Abstract

This extended abstract summarises our recent investigation on RDFS-based Knowledge Graphs (RKGs)*. Inspired by previous work on equipping DLs with a semantics adequate for metamodeling, we provide a formal semantics for RKGs based on classical logic. We show that, surprisingly, under the newly defined semantics, RKGs do not admit, in general, a canonical model. Also, we introduce the notions of definite and indefinite RKGs and show that being definite is both a sufficient and necessary condition for an RKG to admit a canonical model, thus singling out the source of incompleteness that causes the lack of a canonical model for indefinite RKGs. Finally, we characterize the complexity of the query answering problem for both definite and indefinite RKGs.

Keywords

knowledge graphs, metamodeling semantics, RDFS

The rapid advancements in Artificial Intelligence (AI) in recent years have significantly increased interest in Knowledge Graphs (KGs) [1], both in academia and industry. KGs have become foundational to AI systems due to their ability to model complex domains by structuring entities and their interrelations in a semantically rich format. Several classes of frameworks for managing KGs exist [1]. One consists of KGs as plain graph databases, queried via pattern-matching languages, such as property graphs [2, 3]. These, however, suffer from semantic limitations, as the graph itself represents a single model, preventing reasoning tasks such as consistency checking or deductive knowledge inference. In this context a KG does not allow for the execution of reasoning tasks based on the graph semantics, as checking consistency of the graph or completing it by means of new knowledge inferable via deduction. A different approach is the one where KGs are represented as Description Logic (DL) knowledge bases (or ontologies) [4, 5, 6]. Despite strongly enriching the reasoning capabilities provided by such kind of KGs, this approach still presents some limitations derived from the lack of metamodeling capabilities. A common solution to this limitation lies in the use of *punning*, a syntactic workaround allowing for the occurrence of the same syntactic element in positions representing different roles, but with punning, different occurrences actually represent different semantic elements. While a metamodeling semantics has been proposed for the ontology language OWL2 QL [7], to the best of our knowledge no system exists implementing such framework. Another approach to manage KGs is represented by RDFS, a framework derived from the Semantic Web domain.

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland

delfino@diag.uniroma1.it (R. M. Delfino); lenzerini@diag.uniroma1.it (M. Lenzerini); poggi@diag.uniroma1.it (A. Poggi)

0000-0002-5492-5290 (R. M. Delfino); 0000-0003-2875-6187 (M. Lenzerini); 0000-0002-4030-3458 (A. Poggi)

* The present extended abstract is based on a paper accepted at ECAI 2025.

RDFS allows for the representation of both intensional and extensional knowledge expressed as triples of the form $\langle \mathtt{s\ \ p\ \ o} \rangle$, each representing an edge p connecting a node s to a node o. By including a vocabulary of special symbols provided with semantics, RDFS allows to capture the meaning of some fundamental relationship types existing between the entities of the graph, such as membership assertion or subset relations between classes or properties. Despite its long-standing presence and widespread adoption in knowledge graph management, several inherent limitations of RDFS have been identified, which we argue require to be seriously addressed. One objection that arose with the original framework was based on the incompleteness deriving from the so-called entailment rules, i.e., the rules for entailment the RDFS semantics is based on [8, 9]. More importantly, several works have pointed out severe drawbacks of the RDFS standard semantics, also referred to as *intensional semantics*, (see, e.g., [10]). In particular, such semantics fails to define sets in terms of their extensions, thus not being able to fully capture the set-theoretic notions underlying basic constructs such as the subset relation (e.g., the one corresponding to the `subClassOf` construct). This affects the significance of the represented knowledge and the compatibility with most widely used, logic-based knowledge representation languages, such as OWL. To address such a limitation, RDFS was provided with a “non-normative" semantics, called *extensional* [11], expressed through an additional set of entailment rules aiming to capture the classical logic semantics. Also, in [10], the authors propose a proof-theoretic approach for RDFS entailment based on the extensional semantics and on a set of entailment rules which slightly extends the one of the “non-normative" RDFS semantics. Finally, most of the works addressing the problem of answering queries over RDFS KGs resort to the RDFS entailment regime [12, 13, 14, 15], according to which, existential variables within queries are not treated as in first-order classical logic. Indeed, such semantics requires the existence of a binding of each such variable to the same domain object in every model. This is clearly a limitation, compared to the classical logic semantics which looks for the existence of a binding in every model, possibly accepting different bindings in different models. The only work addressing classical logic query answering over RDFS is [11], which analyses both semantics model-theoretically and provides complexity results for graph entailment. However, the query answering problem under the extensional semantics remains open. Based on the above considerations, the main contribution of this workis to develop algorithms and complexity analyses for query answering over RDFS KGs under classical logic. More precisely, we focus on a specific type of graphs, called *RKGs*, that capture the core of RDFS. Inspired by [7], we propose a logic-based metamodeling semantics for RKGs and show that, under such semantics, RKGs do not admit, in general, a canonical model. We introduce the notions of definite and indefinite RKGs and show that being definite is both a sufficient and necessary condition for an RKG to admit a canonical model, thus singling out the source of incompleteness that causes the lack of a canonical model for indefinite RKGs. Finally, we characterize the combined complexity (where both the RKG and the query Q are provided in input) of query answering for both definite and indefinite RKGs.

Definition 1. *Given a set of IRIs $\mathbb{I}$, a set of IRIs $\mathbb{R}$ = {type, subClassOf, subPropertyOf, domain, range, Resource, Class, Property} $\subseteq \mathbb{I}$, and a set of symbols $\mathbb{B}$ denoting blank nodes (we assume the symbols in $\mathbb{B}$ to start with “_"), an RKG G is a set of triples of the form $\langle s\ \ p\ \ o \rangle$, where s, p, o*

https://www.w3.org/TR/rdf-mt/

$\in \mathbb{I} \cup \mathbb{B}$.

Note that RKGs comprise a subset of the RDFS vocabulary, do not contain any literal, and possibly include blank nodes in predicate position.

Example 1. *The following set of triples represents a valid RKG:* $\{\langle$*_b*$_1$ *teachesTo Alice*$\rangle$, $\langle$*teachesTo range Student*$\rangle$, $\langle$*teachesTo domain Professor*$\rangle$, $\langle$*Professor subClassOf Person*$\rangle$, $\langle$*Professor type FacultyRole*$\rangle$, $\langle$*FacultyRole _b*$_2$ *Role*$\rangle\}$.

The semantics of RKGs is defined by resorting to the notion of interpretation, where an interpretation $\mathcal{I}$ for an RKG G is a pair $\langle \mathcal{W}, \cdot^{\mathcal{I}} \rangle$, where $\mathcal{W}$ is called interpretation structure of $\mathcal{I}$, and $\cdot^{\mathcal{I}}$ is the interpretation function of $\mathcal{I}$. In particular, $\mathcal{W}$ is a triple $\langle \Delta, \cdot^{C}, \cdot^{P} \rangle$, such that Δ is a non-empty set of objects, called the *domain* of $\mathcal{I}$, while $\cdot^{C}$ and $\cdot^{P}$ are partial functions. Intuitively, $\cdot^{C}$ (resp., $\cdot^{P}$) is defined for those domain objects that play the role of class (resp., property) in $\mathcal{W}$ and determines their extension as a class (resp., property). The interpretation function maps every IRI appearing in a graph into an object in Δ. All symbols from the set $\mathbb{R}$ are interpreted according to their intended set-theory meaning. As an example, $(\texttt{subClassOf}^{\mathcal{I}})^{C} = \{(o_1, o_2) | o_1, o_2 \in \Delta \text{ and } o_1^{C} \subseteq o_2^{C}\}$. Blank nodes are dealt with by means of an assignment function $\nu_{\mathcal{I}}$ such that, if x is an IRI, then $\nu_{\mathcal{I}}(x) = x^{\mathcal{I}}$, while if x is a blank node, then $\nu_{\mathcal{I}}(x) = o$ (for some $o \in \Delta$). We say that an interpretation $\mathcal{I}$ for G satisfies a triple $\langle$ a b c $\rangle$ under $\nu_{\mathcal{I}}$, denoted $(\mathcal{I}, \nu_{\mathcal{I}}) \models \langle\ \texttt{a b c}\ \rangle$, if $(\nu_{\mathcal{I}}(\mathrm{a}), \nu_{\mathcal{I}}(\mathrm{c})) \in \nu_{\mathcal{I}}(\mathrm{b})^{P}$. If an interpretation $\mathcal{I}$ satisfies every triple of G, denoted $\mathcal{I} \models G$, $\mathcal{I}$ is called a *model* of G. We denote $Mod(G)$ the set of models of G.

Definition 2. *Let* $\mathcal{I}$ *be an interpretation for an RKG* G *with domain* Δ *and* Q *be a query containing the set of IRIs* $\mathbb{I}^{Q}$ *and the set of blank nodes* $\mathbb{B}^{Q}$*. A query homomorphism from* Q *to* $\mathcal{I}$ *is a total function* $\Psi : \mathbb{I}^{Q} \cup \mathbb{B}^{Q} \rightarrow \Delta$*, such that for every* $s \in \mathbb{I}^{Q}$, $\Psi(s) = s^{\mathcal{I}}$*, and for every* $\langle s_1\ s_2\ s_3 \rangle \in Q$, $(\Psi(s_1), \Psi(s_3)) \in \Psi(s_2)^{P}$.

Intuitively, $\mathcal{I} \models Q$ if and only if there exists a query homomorphism from Q to $\mathcal{I}$.

Definition 3. *An RKG* G *entails a query* Q*, denoted* $G \models Q$*, if there exists a query homomorphism from* Q *to* $\mathcal{I}$ *for every* $\mathcal{I} \in Mod(G)$,

Analogously to what we have done in Definition 2, it is possible to define the notion of homomorphism from an interpretation $\mathcal{I}$ to an interpretation $\mathcal{J}$. Also, as usual, we say that a model $\mathcal{I}$ is a *canonical model of an RKG* G if, for every model $\mathcal{J}$ of G, there exists a homomorphism from $\mathcal{I}$ to $\mathcal{J}$. Despite RDFS is generally considered a "lightweight" language, and such languages typically admit a canonical model that can be exploited for answering queries and for other reasoning tasks, we have the following surprising result.

Proposition 1. *There exists an RKG* G *such that no interpretation of* G *is a canonical model.*

To illustrate the significance of the above result, consider the RKG $G = \{\langle \texttt{a R b} \rangle, \langle \texttt{b R a} \rangle, \langle \texttt{t type b} \rangle, \langle \texttt{a type Class} \rangle\}$, and the query $Q : \{\langle \texttt{\_x R \_y} \rangle, \langle \texttt{\_z type \_y} \rangle, \langle \texttt{\_x subClassOf b} \rangle\}$. On can verify that G entails Q, since in every model of G it is possible to find an assignment satisfying the query. In particular, in every model of G where the class

a is empty (which can be codified as a being a subclass of every class in G), the assignment $\{\_x \leftarrow a^{\mathcal{I}}, \_y \leftarrow b^{\mathcal{I}}, \_z \leftarrow t^{\mathcal{I}}\}$ makes the query true, while in all models where a is non-empty, the assignment $\{\_x \leftarrow b^{\mathcal{I}}, \_y \leftarrow a^{\mathcal{I}}, \_z \leftarrow o^{\mathcal{I}}\}$, with o being any instance of a, does it. This shows that we have to *reason by cases*, since there exists no assignment for the variables x, y, z which makes the query true in every model of G. Note, indeed, that Q would not be entailed by G if we adopted the standard SPARQL semantics based on the RDFS entailment regime.

The above case derives from a form of indefiniteness inherent to some RKGs. We singled out the source of such indefiniteness, and we defined two disjoint classes of RKGs, characterized by different properties. Intuitively, given an RKG G and a class a in G, we say that a is *definite* if either it contains instances in every model of G or it is a subset of every class in G. A class that is not *definite* is called *indefinite*. An analogous definition holds for definite and indefinite properties. A graph containing indefinite elements is an indefinite RKG.

Answering queries posed over definite RKGs can be done by means of the so-called *chase procedure* [16], which can be applied to RKGs and which allows to obtain a structure from which it is possible to obtain a canonical model for the given RKG, similarly to what happens for several lightweight ontology languages [17].

Proposition 2. *Query entailment in definite RKGs can be done in polynomial time.*

Since indefinite RKGs do not admit a canonical model (see Proposition 1), for such RKGs query entailment requires using techniques based on reasoning by cases. The algorithm that we propose works as follows. Given an indefinite RKG G, it guesses a set of indefinite classes and properties, and it generates a new RKG G' (called a *completion* of G) obtained from G by making each guessed class and property definite by providing them with new instances, while the non-guessed ones are made definite by adding triples that make them subsets of every class and every property, respectively. By guessing all possible combinations of indefinite classes and properties, query entailment for an RKG G and a query Q can be solved by checking if there exists at least a completion that makes the query false. If that is the case, then we can conclude that $G \not\models Q$. On the contrary, if such a graph does not exist, then we can conclude that $G \models Q$. Thus, it is possible to solve the query entailment problem for general RKGs in Π_2^p with respect to the size of the entire input. By means of a reduction from the satisfiability of 2-QBF formulas problem, we also provide a matching lower bound for the query entailment problem.

Theorem 1. *Query entailment in RKGs is Π_2^p-complete in combined complexity.*

Future developments of the framework proposed in this paper ideally involve the use of epistemic logic, as a tool to capture different interpretations for the semantics of queries [18], and the extension of both graph and query language with forms of negations [19, 20, 21].

Acknowledgments

This work has been supported by MUR under the PNRR project FAIR (PE0000013).
This work has been carried out while Roberto Maria Delfino was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome.

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Computing Surveys (Csur) 54 (2021) 1–37.

[2] J. L. Larriba-Pey, N. Martínez-Bazán, D. Domínguez-Sal, Introduction to graph databases, in: Reasoning Web International Summer School, Springer, 2014, pp. 171–194.

[3] R. Angles, C. Gutierrez, Survey of graph database models, ACM Computing Surveys (CSUR) 40 (2008) 1–39.

[4] M. Krötzsch, Ontologies for knowledge graphs?, in: A. Artale, B. Glimm, R. Kontchakov (Eds.), Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017, volume 1879 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017. URL: https://ceur-ws.org/Vol-1879/invited2.pdf.

[5] D. Calvanese, G. De 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.

[6] 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 (2016) 471–487.

[7] M. Lenzerini, L. Lepore, A. Poggi, Metamodeling and metaquerying in OWL 2 QL, Artif. Intell. 292 (2021) 103432.

[8] D. Marin, A formalization of RDF, Technical Report, Universidad de Chile, 2004.

[9] H. J. ter Horst, Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary, Web Semant. 3 (2005) 79–115. URL: https://doi.org/10.1016/j.websem.2005.06.001. doi:10.1016/j.websem.2005.06.001.

[10] E. Franconi, C. Gutierrez, A. Mosca, G. Pirrò, R. Rosati, The logic of extensional RDFS, in: H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, K. Janowicz (Eds.), The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I, volume 8218 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 101–116. URL: https://doi.org/10.1007/978-3-642-41335-3_7. doi:10.1007/978-3-642-41335-3_7.

[11] J. De Bruijn, S. Heymans, Logical foundations of (e) rdf (s): Complexity and reasoning, in: International Semantic Web Conference, Springer, 2007, pp. 86–99.

[12] S. Munoz, J. Pérez, C. Gutierrez, Simple and efficient minimal RDFS, Journal of web semantics 7 (2009) 220–234.

[13] F. Goasdoué, I. Manolescu, A. Roatiş, Efficient query answering against dynamic RDF databases, in: Proceedings of the 16th International Conference on Extending Database Technology, 2013, pp. 299–310.

[14] M. W. Chekol, J. Euzenat, P. Genevès, N. Layaïda, SPARQL query containment under RDFS entailment regime, in: International Joint Conference on Automated Reasoning, Springer, 2012, pp. 134–148.

[15] M. Buron, F. Goasdoué, I. Manolescu, M.-L. Mugnier, Reformulation-based query answering for RDF graphs with RDFS ontologies, in: The Semantic Web: 16th International

Conference, ESWC 2019, Portorož, Slovenia, June 2–6, 2019, Proceedings 16, Springer, 2019, pp. 19–35.

[16] S. Greco, C. Molinaro, F. Spezzano, Incomplete Data and Data Dependencies in Relational Databases, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2012. URL: https://doi.org/10.2200/S00435ED1V01Y201207DTM029. doi:10.2200/S00435ED1V01Y201207DTM029.

[17] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family, Journal of Automated reasoning 39 (2007) 385–429.

[18] R. Fagin, J. Y. Halpern, Y. Moses, M. Vardi, Reasoning about knowledge, MIT press, 2004.

[19] V. Gutiérrez-Basulto, Y. A. Ibáñez-García, R. Kontchakov, E. V. Kostylev, Queries with negation and inequalities over lightweight ontologies 35 (2015) 184–202.

[20] G. Cima, M. Lenzerini, A. Poggi, Answering conjunctive queries with inequalities in $\mathit{DL\text{-}Lite}_R$, 2020, pp. 2782–2789.

[21] G. Cima, M. Console, R. M. Delfino, M. Lenzerini, A. Poggi, Answering Conjunctive Queries with Safe Negation and Inequalities over RDFS Knowledge Bases, Proceedings of the AAAI Conference on Artificial Intelligence 39 (2025) 14824–14831. URL: https://ojs.aaai.org/index.php/AAAI/article/view/33625. doi:10.1609/aaai.v39i14.33625.