

On the Need of a Formal Meta-modeling Semantics for Knowledge Graphs

Roberto M. Delfino*, Maurizio Lenzerini and Antonella Poggi

Sapienza University of Rome, 00185 Rome, Italy

Abstract

We discuss several limitations of current formalizations of Knowledge Graphs, considering in particular the central role they play in data preparation and in neuro-symbolic AI. We briefly discuss the characteristics of our proposal for a comprehensive framework based on formal logic whose goal is to both overcome these deficiencies and provide suitable tools for accommodating various extensions to the modeling features and the querying capabilities of AI systems based on Knowledge Graphs.

Keywords

Knowledge Graphs, Semantics, Metamodeling

1. Introduction

Recent years have seen an increasing interest in both research and industry in two fundamental aspects involving Artificial Intelligence (AI) and Data Management at different levels and several works emphasize the role of Knowledge Graphs (KGs) in both of them.

The first one is related to *Data Preparation*, i.e., the process of collecting, cleaning, transforming, and organizing raw data into a format that is suitable for analysis or use in machine learning. It is well-known that, although the goal of data preparation is clear, finding an effective solution is much more complex. While there are numerous approaches available, their success varies widely. There is a general consensus, however, that a more long-term fix can be found in standardization. By adopting uniform data representations, terminologies, and vocabularies at the time of data modeling, organizations can drastically cut down on the need for repetitive actions as new sources or requirements emerge. Toward this goal, using KGs simplify data gathering and integration and offer significant advantages for preparing data and utilizing them for actionable insights.

The second one is *Neuro-symbolic AI* [1, 2], a set of techniques aiming at combining two different and complementary paradigms in AI in order to overcome the limitations of either approach. While the capabilities of ML-based approaches in many cases exceeded expectations, they still present some drawbacks, mainly related to the possibility of properly interpreting data, taking advantage of deductive reasoning and providing forms of explainability. Such capabilities could be obtained through knowledge-driven techniques from symbolic AI, e.g., those based on formal languages and logics. KGs represent a domain by means of a comprehensive structure including both intensional and extensional knowledge. Once provided with the right semantics, KGs seem an ideal tool to perform both inductive and deductive reasoning tasks, which is a prerequisite for integrating symbolic reasoning and neural network learning.

Although several frameworks exist that are used to manage knowledge graphs, such as simple graph structures [3], graphs databases equipped with a query language [4][5], RDFS graphs [6], or DL-based knowledge bases [7], *we argue that each of these approaches exhibits distinct limitations that should be addressed to fully harness the modeling and reasoning capabilities of knowledge graphs, so as to exploit their full potential in both data preparation and neuro-symbolic systems.*

AIxIA 2024 Discussion Papers - 23rd International Conference of the Italian Association for Artificial Intelligence, Bolzano, Italy, November 25–28, 2024

*Corresponding author.

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

🌐 https://www.diag.uniroma1.it/users/roberto-maria_delfino (R. M. Delfino)

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



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The goal of this position paper is to discuss which are the current approaches adopted to represent and manage KGs, together with specific arguments about how each of them lacks in properly managing different semantic-related aspects (section 2) and to illustrate the basic characteristics of our approach, based on formal logic, aimed to overcome the above-mentioned limitations (section 3).

2. Current Approaches

Several different methods exist to represent KGs. Among those, we will consider three of them: graph databases, RDFS graphs and DL-based knowledge bases.

Graph databases In this approach a KG is seen simply as a database structured as a directed graph, possibly equipped with additional syntactic elements, such as labels and properties associated to either nodes or edges (in this case they are called property graphs [4]). Nodes represent domain elements, while edges represent relationships existing between them. Graph databases are typically equipped with specialized query languages, with Cypher being the most widely used for property graphs [5]. One of the most evident limitations of graph databases comes from the fact that they do not come with rich semantic features: in most cases they are plain graph databases carrying exclusively extensional knowledge, while not specifying any intensional knowledge describing relations between classes of objects or relationships. This makes it hard to model sophisticated domain features, to check for consistency or to apply deductive reasoning on the graph. The corresponding query languages are based on the concept of pattern-matching, i.e., a query takes into account a single logical model, corresponding to the graph itself, and looks for bindings to be used for matching patterns specified in the query. As we will see later, this can be a serious limitation in terms of semantic characterization for query answering.

RDFS Another very common formalism used to represent knowledge graphs is RDFS [8]. An RDFS graph is a set of triples of the form $\langle s \ p \ o \rangle$ where p represents a relationship (called predicate) existing between a subject s and an object o . In other words, every triple can be seen as a graph fragment, where s and o represent nodes and p represents a directed edge from s to o . Differently from plain graph databases, RDFS is equipped with a vocabulary of reserved symbols with an assigned semantics, which can be used to represent intensional knowledge in terms of classes, predicates, ISA relationships, class and property membership. Despite this, the standard semantics of RDFS falls short in adequately capturing some semantic aspects (see [9]), and, moreover, it lacks any form of incomplete information in domain modeling. We will come back later to these issues through an example.

DL knowledge bases An alternative approach is that of DL knowledge bases or ontologies. A DL knowledge base consists in a set of axioms describing both intensional and extensional knowledge, written in a suitable Description Logic language. DL ontologies have been extensively studied and they typically come with formal and precise semantics, which would overcome some of the problems of property graphs and RDFS graphs. Still, such systems based on the classical approach to DLs are not capable of properly capturing metamodeling flavours, i.e., semantic aspects involving domain elements which can simultaneously play the role of class and individual, or of class and property. Also, similarly to RDFS-based tools, such systems often adopt a semantics based on punning, according to which two occurrences of the same syntactic domain element in different positions (e.g. class position and individual position) are treated as if they refer to different elements. It has been observed that this does not constitute a proper metamodeling semantics [10].

3. Our approach

With the goal of overcoming the above-mentioned issues, we are working on the definition of a general semantic framework for KGs. The basic idea of the framework is to assign a well-founded meaning,

based on formal logic, to any graph given in input and to provide a solid basis, still based on logic, for interpreting queries over it.

Suppose we are given the graph in Figure 1, and assume we want to know if there are x, y, z such that x is connected to y by `coWorkerOf`, z is of type y and x is a subclass of `Developer`. With the

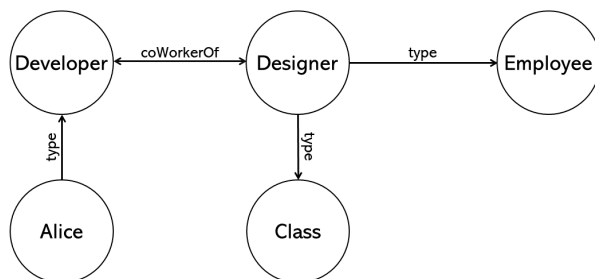


Figure 1: Example of knowledge graph. The label `type` has the semantics of `rdf:type`, i.e., the element denoted by the source node is an instance of the class denoted by the target node.

graph database approach, the query gets the answer "false", simply because there is no explicit reference to the notion of subclass in the graph.

Similarly, with the RDFS approach, the SPARQL query corresponding to the above question gets the answer "false", because, even considering the correct inference rules on `type` and `subclassOf`, there is no binding for the variables x, y, z such that the corresponding pattern appears in the graph.

Finally, with the typical first-order semantics of DLs, and, more generally, in all the first-order semantics based on punning, the answer gets again the answer "false", because no binding exists assigning to y the same element, since no first-order model exists where the same element plays both the role of an object (in the query, y plays this role since it is connected to x by `coWorkerOf`) and of a predicate (in the query y plays this role since z is of type y).

However, with a formal semantics based on logic (and, therefore, on the open-world assumption) and taking meta-modeling seriously, the correct answer is "true": indeed, in the models where the class `Designer` is empty, the query is satisfied by the binding $x \rightarrow \text{Designer}$, $y \rightarrow \text{Developer}$, and $z \rightarrow \text{Alice}$, whereas in the models where `Designer` is not empty (i.e., it has at least one instance, say c), the query is clearly satisfied by the binding $x \rightarrow \text{Developer}$, $y \rightarrow \text{Designer}$ and $z \rightarrow c$.

Notice that a proposal of Metamodeling Semantics has been put forward for OWL 2 QL ontologies [10], where the authors discuss in detail why the *punning* offered by the *Direct Semantics* of OWL 2 QL, does not address the problem of meta-modeling in a correct way. Such a meta-modeling semantics is indeed the basis of our approach. More specifically, our short term goals are as follows.

- We aim at providing a comprehensive framework for a correct semantic characterization of KGs, based on meta-modeling. The framework should be able to specify suitable constraints for obtaining the formalization of specific forms of graphs. For example, the framework should specify which are the constraints on input graphs and queries that correspond to the interpretation of the graph as a property graph processable with Cypher.
- We aim at capturing several forms of incomplete information arising in the different variants of KGs. A notable example is RDFS, where many constructs (e.g., `rdf:Statement`) give raise to incompleteness once interpreted correctly. Current approaches to RDFS just ignore such constructs.
- We aim at extending both the modeling constructs of KGs and the query languages with more powerful features, such as negation, inequality, integrity constraints and epistemic queries. We believe that the last feature is particularly important in order to nicely capture weak forms of reasoning corresponding to the *Direct Semantics* of SPARQL, and integrate them with more powerful forms of reasoning.

Acknowledgments

This work has been supported by MUR under the PNRR project FAIR (PE0000013).

This work has been carried out while Roberto M. Delfino was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome.

References

- [1] P. Hitzler, A. Eberhart, M. Ebrahimi, M. K. Sarker, L. Zhou, Neuro-symbolic approaches in artificial intelligence, *National Science Review* 9 (2022).
- [2] A. Oltramari, J. Francis, C. Henson, K. Ma, R. Wickramarachchi, Neuro-symbolic architectures for context understanding, in: *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges*, IOS Press, 2020, pp. 143–160.
- [3] F. N. Stokman, P. H. de Vries, Structuring knowledge in a graph, in: *Human-Computer Interaction: Psychonomic Aspects*, Springer, 1988, pp. 186–206.
- [4] R. Angles, The property graph database model., in: *AMW*, 2018.
- [5] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, A. Taylor, Cypher: An evolving query language for property graphs, in: *Proceedings of the 2018 international conference on management of data*, 2018, pp. 1433–1445.
- [6] W. Ali, M. Saleem, B. Yao, A. Hogan, A.-C. N. Ngomo, A survey of rdf stores & sparql engines for querying knowledge graphs, *The VLDB Journal* (2022) 1–26.
- [7] F. Baader, I. Horrocks, C. Lutz, U. Sattler, *An Introduction to Description Logic*, Cambridge University Press, 2017.
- [8] W3C, RDF schema 1.1, <https://www.w3.org/TR/rdf-schema/>, 2014.
- [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] M. Lenzerini, L. Lepore, A. Poggi, Metamodeling and metaquerying in owl 2 ql, *Artificial Intelligence* 292 (2021) 103432.