Understanding KGs

By: Michael McCain
Shreyas Casturi
Susan Shrestha
Chris Davis Jaldi

Knowledge Graphs: Data Graphs

Takeaways

- Graph models store connections between data points, making them ideal for representing relationships.
- Different graph models (directed edge-labeled, property graphs, etc.) serve different use cases and complexity levels.
- Graph patterns act like templates to match and extract specific subgraphs from your data.

Knowledge Graphs: Data Graphs

Takeaways

- Query results can be manipulated using relational operations like projection, selection, and joins.
- Path expressions enable navigation through multiple connections, finding relationships that span several hops.
- Graph databases like Neo4j implement these concepts for real-world applications.

Knowledge Graphs: Data Graphs

Q & A

If KGs can use different graph models, what requirements determine which one to choose?

- **Directed Edge Labelled Graph:** Prioritize strategic logical definition and advanced deductive reasoning by maintaining to formal semantics and open W3C standards. [8]
- **Property Graph:** Prioritize tactical analytical speed and high performance for managing big data and complex graph traversal workloads. [8]

Knowledge Graphs: Data Graphs

Q & A

Discuss more on Heterogeneous Graph.

- A **Heterogeneous Graph (HG)**, also known as a **Heterogeneous Information Network (HIN)**, is a specialized network model designed to reflect the non-uniform complexity of real-world data. [9]
- It rigorously defines multiple distinct types of nodes and edges, preserving critical semantics often lost when modeling complex systems. [9]

Knowledge Graphs: Data Graphs

Q & A

Discuss more on Heterogeneous Graph.

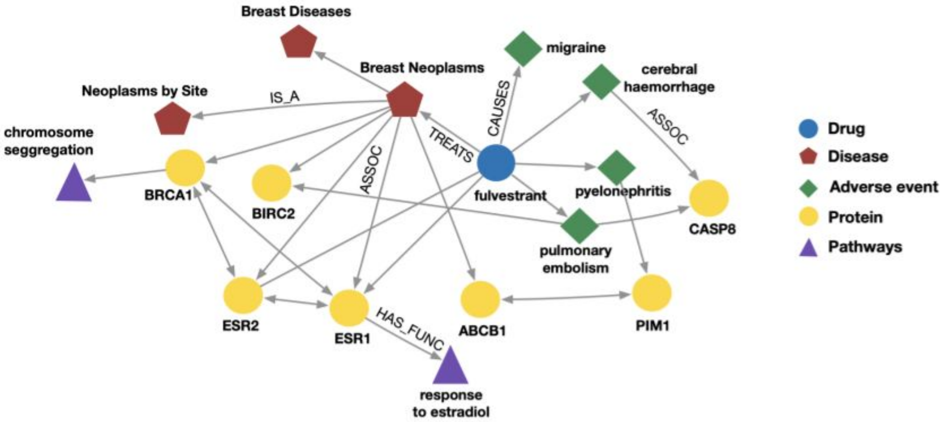


Fig: Example of Heterogeneous Graph

Knowledge Graphs: Data Graphs

Q & A

In what way does the modern KG integrate the best aspects of existing models?

- **Logical Backbone (RDF):** Use the RDF model to enforce definitive facts, standards, and deep logical reasoning. [8]
- **Speed and Analytics (Property Graph):** Use the Property Graph model for high-speed performance on analytical queries and graph traversals. [8]
- **AI/ML Focus (Heterogeneous Graph):** Integrate specialized Heterogeneous Graph structures when modeling complex, multi-typed relationships necessary for advanced AI applications. [9]

Knowledge Graphs: Schema, Identity, Context

Takeaways

- **Shapes** are used to define a **validation graph**. These shapes are a set of nodes that have been selected for specific constraints.
- **Semantics**, **validation**, and **emergent schemata** together support reasoning, constraint validation, and automatic structure discovery within knowledge graphs.
- **Lexicalisation** through the use of **identifiers** with labels is a great way to ground a nodes identity and allow for cross referencing in other texts.

Knowledge Graphs: Schema, Identity, Context

Takeaways

- When representing data of many varieties on larger scales the default choice in semantic representation is **OWA**.
- Contextual information scoping like **provenance** and **temporal scope** complements schema and identity to ensure that knowledge remains meaningful and accurate across different context specific situations.

Knowledge Graphs: Schema, Identity, Context

Q & A

What are the similarities and differences between reification and Higher-arity representation?

- **Reification**: Making statements about statements. Useful for including provenance data about a thing. The Statement becomes a node [2, 3]. Used for a single triple.

rdf: Statement

rdf: subject rdf: predicate rdf: object

Fig: A simple example of what can be contained within an rdf: Statement.

```
:statement1 a rdf: Statement ;
        rdf: subject     :Subject ;
        rdf: predicate  :Predicate ;
        rdf: object       :Object ;
:statement1 :statedBy :Person1 .
```

Fig: How a reification would look in TTL.

Knowledge Graphs: Schema, Identity, Context

Q & A

What are the similarities and differences between reification and Higher-arity representation?

- **Higher-arity**: Making a statement about statements, but for an entire graph. These graphs can be either subjects or objects of RDF triples [1, 2, 3].

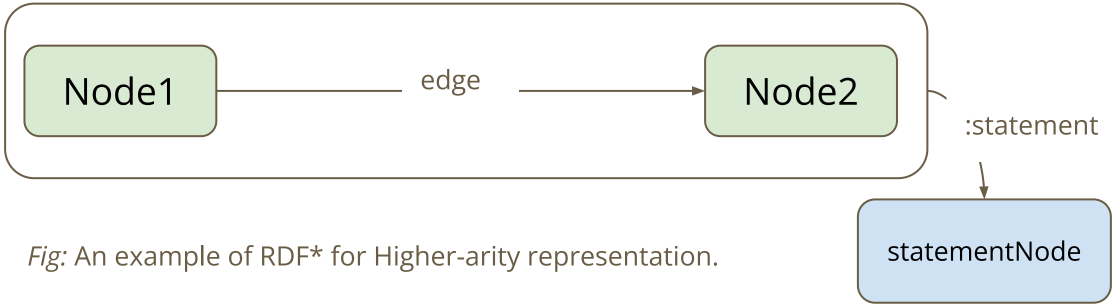


Fig: An example of RDF* for Higher-arity representation.

<< :node1 :edge :node2 >> :statement :statementNode .

Fig: How a RDF* statement would look in TTL.

Knowledge Graphs: Schema, Identity, Context

Q & A

What are the differences between open (OWA) and closed world assumptions (CWA)?

- CWA:
  - Anything unknown is <u>false</u>.
  - In other words, if a **fact** does not exist in the **KB**, then it false [3].
- OWA:
  - Anything unknown is simply that and is **NOT** assumed to be false [3].

Knowledge Graphs: Schema, Identity, Context

Q & A

Can you more clearly define a quotient graph?

- A **quotient graph** is a summary of a **data graph** which shows the high level structure by merging (**partitioning**) nodes from the original data graph based on an equivalent relationship [3].

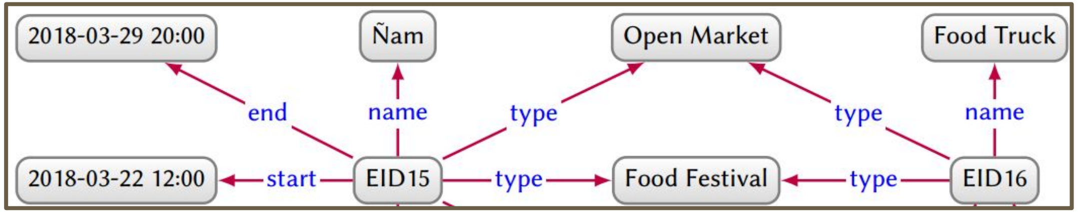


Fig: Directed edge-labelled graph describing events and their venues [3].

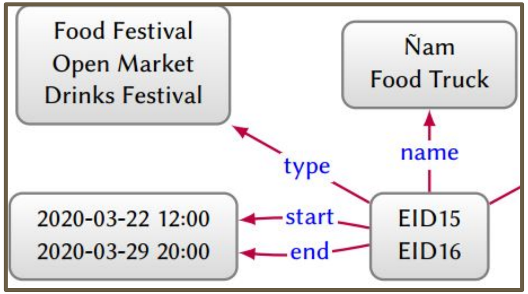
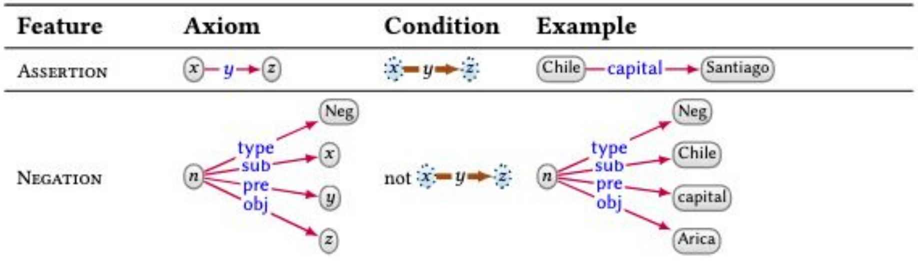


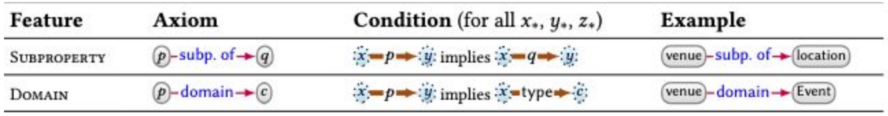
Fig: A resulting quotient graph after partitioning [3].

Knowledge Graphs: Deductive Knowledge

Takeaways

- **<u>Q: Can machines deduce new knowledge from a data graph?</u>**
- Machines can "deduce" by producing conclusions from a set of premises (*entailment regimes)*
- Machines require
  - *ontologies* (a schema of knowledge that can be mapped to a graph)
  - *rules for interpretations* (ways to understand the data in the context of the real world)





Figures: Relationships between axioms (properties) and semantic conditions (meanings) from Hogan et al.

Knowledge Graphs: Deductive Knowledge

Takeaways

- **Semantic conditions** allow for *entailments* on graphs
- Whether one graph can be entailed by another is an *undecidable problem* – approx. solutions below:
  - Sol 1: halt on any input ontology but miss entailments
  - Sol 2: always halt on restricted feature ontologies
  - Sol 3: Only return correct answers for an ontology, but may never halt on certain inputs
- There are two approaches to entailment:
  - Adding If-Then/If-Then-Else conditional Rules
  - Using Description Logics to assert axioms about individuals, classes (groups/concepts) and properties (roles)

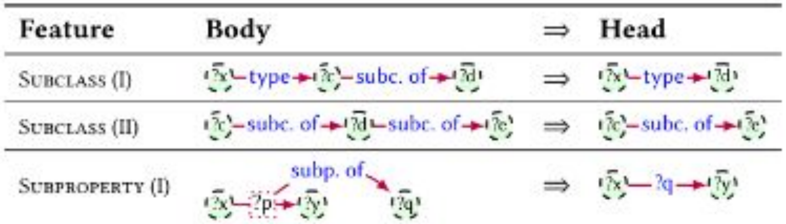


Figure: Example of IF-THEN rules for specific properties: IF (body) THEN (head), from Hogan et al.

Knowledge Graphs: Deductive Knowledge

Q & A

What is meant by "Expressive Deduction Logics" in the last paragraph on page 32?

- *Expressive* in that context refers to DLs that have different constructs which can "express" or capture certain properties better. Some general examples below, from [4]:
  - SROIQ is a well-known DL with certain axiomatic and constructor restrictions
  - ALC is a more restricted SROIQ-based DL, which lacks even more axioms and has more constraints
  - EL is another DL family that has unlimited existential quantifiers
- These DLs allow us to capture certain relationships, but have tradeoffs on computability and decidability

Knowledge Graphs: Deductive Knowledge

Q & A

How can we balance between more expressive ontologies – which allow richer reasoning – and computational tractability in real-world?

- Ultimately, this comes down to the use case and the reasoners we use.
- Do we want
  - fast entailments which are generally correct,
  - entailments that are always correct but very slow to calculate, etc...
- Hogan et al. state: "[the option of always getting the correct entailment]... only accept[ing] [restricted] input ontologies... may be a better choice in [medical] domains... where missing entailments may have undesirable outcomes", as an example

Knowledge Graphs: Deductive Knowledge

Q & A

How do SPARQL Inferencing Notation (SPIN) and Description Logic work for real-world deductive reasoning? Can they work together?

- NOTE: SPIN has been deprecated in favor of SHACL.
- SPARQL is a query language for RDF and OWL ontologies, per [6], and SPIN allows one to express SPARQL queries as RDF triples, per [5].
- DLs, on the other hand, are a different type of ontology modeling language [4], with links to OWL Web Ontology Language. DLs give rise to Direct Semantics, whereas there are RDF semantics (which make reasoning undecidable)
- DLs do have Query Languages, such as DIG protocol and nRQL queries [6], but there is a problem in combining SPARQL/SPIN with OWL-DL-based ontologies, as seen by [6]

Some efforts were made to bridge SPARQL and DL families.

- The authors in [6] try to create a SPARQL-like QL for OWL-DL ontologies
- The paper "Using Description Logics for RDF Constraint Checking and Closed-World Recognition" [7] by Patel-Schneider may provide more details, where DL axioms can be used for constraint checking on RDF graphs, and constraint checking/closed-world recognition can be translated to SPARQL queries.

Knowledge Graphs: Creation and Enrichment

Takeaways

- Ontology engineering as a downstream step
  - Comes after data ingestion and processing to ensure entities, relations, and schemas are structured correct.
- Two modes of ontology creation:
  - Ontology learning (automatic ML/NLP-driven) - scales well but needs careful validation to not introduce errors
  - Human-based construction - provides domain expertise, precision, and curation but expensive

Knowledge Graphs: Creation and Enrichment

Takeaways

- NLP techniques at the core of building and enriching KGs
  - Named Entity Recognition (NER); Entity Linking (EL); Relation Extraction (RE)
- Challenges with entity linking:
  - Ambiguous entities requiring disambiguation using contextual ranking
  - Consistent representation of entities with aliases
- SPARQL for enrichment by filtering subsets
- Validation of ontologies (KGs) with CQs

Knowledge Graphs: Creation and Enrichment

Q & A

How do transformers and ANNs perform at ontology learning from text? What bottlenecks exist?

- Transformers (LLMs) show promise wrt extraction of richer semantics [10] but bottlenecks include lack of domain-specific training data, interpretability, and difficulty in aligning extracted patterns with formal ontologies

Among enrichment methods (human collaboration, text/markup extraction, structured sources, leveraging external KGs), which are most impactful at scale?

- Incremental enrichment with structured sources and external KGs scale resource effective but human based extraction might give better quality in data with formal ontology considerations. [11]

Knowledge Graphs: Creation and Enrichment

Q & A

When validating an ontology/KG with results partially incomplete/inaccurate, what's the troubleshooting order?

- Verify the query with CQ - > Check required data existence in KG - > Check for valid ingestion and materialization - > Revisit ontology design and schema wrt to CQs; Verify Key notions [12]

How has ontology learning been applied in practice?

- Ontology based construction is a often time consuming for manual processing and efforts towards automation continue to be explored which mostly includes AI models, with some effort towards Semantic web migration. Some practical examples include DBLP, medical applications etc [10, 13, 14, 15, 16]

References

[1] ISE FIZ Karlsruhe, “Knowledge Graphs - Excursion 1: Reification and RDF*,” YouTube, Oct. 26, 2023. Available: https://www.youtube.com/watch?v=4KGMtC_bLuE.

[2] Dr. Al-Hakam Hamdan, “Reification in RDF,” YouTube, Jun. 29, 2025. Available: https://www.youtube.com/watch?v=ArPP5Iauofc.

[3] A. Hogan et al., “Knowledge Graphs,” ACM Computing Surveys, vol. 54, no. 4, pp. 1–37, May 2022, doi: https://doi.org/10.1145/3447772

[4] M. Krotzsch, F. Simancik, and I. Horrocks, "A Description Logic Primer", arxiv:1201.4089v3, 3 June 2013. https://arxiv.org/pdf/1201.4089

[5]. H. Knublauch, "SPIN In Five Slides", https://www.slideshare.net/slideshow/spin-in-five-slides/5272225

[6]: E. Sirin and B. Parsia, "SPARQL-DL: SPARQL Query for OWL-DL", "OWLED 2007 Workshop on OWL: Experiences and Directions, CEUR Workshop Proceedings, Vol-258. https://ceur-ws.org/Vol-258/paper14.pdf

[7]: P. F. Patel-Schneider, "Using Description Logics for RDF Constraint Checking and Closed-World Recognition", Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015): https://arxiv.org/abs/1411.4156

[8]:DataWalk, "Property Graph vs. RDF?," 2025. [Online]. Available: ..

References

[9]:L. M. C. W. Junwei Su, "BG-HGNN: Toward Scalable and Efficient," 13 3 2024.

[10]: Norouzi, S. S., Barua, A., Christou, A., Gautam, N., Eells, A., Hitzler, P., & Shimizu, C. (2024). Ontology population using llms. arXiv preprint arXiv:2411.01612.

[11]: Freund, M., Schmid, S., Dorsch, R., & Harth, A. (2024, May). FlexRML: a flexible and memory efficient knowledge graph materializer. In European Semantic Web Conference (pp. 40-56). Cham: Springer Nature Switzerland.

[12]: Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

[13]: Martinez-Rodriguez, J. L., López-Arévalo, I., & Rios-Alvarado, A. B. (2018). Openie-based approach for knowledge graph construction from text. Expert Systems with Applications, 113, 339-355.

[14]: Lo, A., Jiang, A. Q., Li, W., & Jamnik, M. (2024). End-to-end ontology learning with large language models. Advances in Neural Information Processing Systems, 37, 87184-87225.

[15]: Shimizu, C., & Hitzler, P. (2025). Accelerating knowledge graph and ontology engineering with large language models. Journal of Web Semantics, 85, 100862.

References

[16]: Mavridis, A., Tegos, S., Anastasiou, C., Papoutsoglou, M., & Meditskos, G. (2025). Large language models for intelligent RDF knowledge graph construction: results from medical ontology mapping. Frontiers in Artificial Intelligence, 8, 1546179