Reasoning over Knowledge Graphs: Motivation, Theory and Practice

Ian Horrocks
Introduction to Knowledge Graphs
Anatomy of a Knowledge Graph

- Architectural Structure
  - Tower
  - City
  - Building
  - Eiffel Tower
  - Shard
  - 324m
  - Paris
  - London
  - 310m
  - France
  - UK
  - EU
Anatomy of a Knowledge Graph
Anatomy of a Knowledge Graph

(subject) London (capital of) UK (member of) EU

(subject) Eiffel Tower (type) Tower

(subject) Paris (location) City

(subject) 324m (height) Architectural Structure

(subject) Shard (type) Building

(subject) 316m (height) Architectural Structure

(subject) Type (kind of) Architectural Structure

(subject) Type (kind of) Eiffel Tower

(subject) Type (kind of) Shard

(subject) Location (type) Architectural Structure

(subject) Location (type) Eiffel Tower

(subject) Location (type) Shard

(subject) Height (type) Architectural Structure

(subject) Height (type) Eiffel Tower

(subject) Height (type) Shard

(subject) Member of (type) Architectural Structure

(subject) Member of (type) Eiffel Tower

(subject) Member of (type) Shard

(subject) Capital of (type) Architectural Structure

(subject) Capital of (type) Eiffel Tower

(subject) Capital of (type) Shard
Anatomy of a Knowledge Graph

Architectural Structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Height</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eiffel Tower</td>
<td>324</td>
<td>Paris</td>
</tr>
<tr>
<td>Shard</td>
<td>310</td>
<td>London</td>
</tr>
</tbody>
</table>

City

<table>
<thead>
<tr>
<th>Name</th>
<th>Capital Of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris</td>
<td>France</td>
</tr>
<tr>
<td>London</td>
<td>UK</td>
</tr>
</tbody>
</table>

Tower

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eiffel Tower</td>
</tr>
</tbody>
</table>

Building

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shard</td>
</tr>
</tbody>
</table>

Member

<table>
<thead>
<tr>
<th>Country</th>
<th>Union</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>EU</td>
</tr>
<tr>
<td>UK</td>
<td>EU</td>
</tr>
</tbody>
</table>
Anatomy of a Knowledge Graph

- Intuitive (e.g., no "foreign keys")
Anatomy of a Knowledge Graph

Architectural Structure

Name | Height | Location
--- | --- | ---
Eiffel Tower | 324 | Paris
Shard | 310 | London

Tower

Name | Building
--- | ---
Eiffel Tower | Shard

City

Name | Capital Of
--- | ---
Paris | France
London | UK

Member

Country | Union
--- | ---
France | EU
UK | EU
Anatomy of a Knowledge Graph

Architectural Structure:
- **Name**: Eiffel Tower
  - **Height**: 324
  - **Location**: Paris
- **Name**: Shard
  - **Height**: 310
  - **Location**: London

City:
- **Name**: Paris
  - **Capital Of**: France
- **Name**: London
  - **Capital Of**: UK

Member:
- **Country**: France
  - **Union**: EU
- **Country**: UK
  - **Union**: EU
Anatomy of a Knowledge Graph

- Intuitive (e.g., no “foreign keys”)
- Data + schema (ontology)
Anatomy of a Knowledge Graph

- Intuitive (e.g., no “foreign keys”)
- Data + schema (ontology)
Anatomy of a Knowledge Graph

- Intuitive (e.g., no “foreign keys”)  
- Data + schema (ontology)  
- URIs not strings
Anatomy of a Knowledge Graph

Architectural Structure

Name | Height | Location
--- | --- | ---
Eiffel Tower | 324 | Paris
Shard | 310 | London

Tower

Name
Eiffel Tower

City

Name
Paris
London

Building

Name
Shard

Member

Country
France
UK

Union
EU
Anatomy of a Knowledge Graph
Anatomy of a Knowledge Graph

- Intuitive (e.g., no “foreign keys”)
- Data + schema (ontology)
- URIs not strings
- Flexible & extensible
Anatomy of a Knowledge Graph

- Intuitive (e.g., no “foreign keys”)
- Data + schema (ontology)
- URIs not strings
- Flexible & extensible
Anatomy of a Knowledge Graph

- Intuitive (e.g., no “foreign keys”)
- **Data + schema (ontology)**
- URIs not strings
- Flexible & extensible
- Other kinds of query
  - navigation
  - similarity & locality
Anatomy of a Knowledge Graph

- Intuitive (e.g., no “foreign keys”)
- Data + schema (ontology)
- URIs not strings
- Flexible & extensible
- Other kinds of query
  - navigation
  - similarity & locality

✗ Views
  - Data integration & restructuring
  - Security
  - Query simplification & optimization
  - ...
Knowledge Graph Semantics
Knowledge Graph Semantics

Why do we need semantics?
- To tell us how to use KG
- E.g., how to answer queries:
  - Architectural Structures with location in the EU?
Knowledge Graph Semantics

(OWL) ontology / conceptual schema

\[ \forall x \text{ Tower}(x) \rightarrow \text{ArchitecturalStructure}(x) \]

\[ \forall x \text{ Building}(x) \rightarrow \text{ArchitecturalStructure}(x) \]
Knowledge Graph Semantics

(OWL) ontology / conceptual schema

∀x Tower(x) → ArchitecturalStructure(x)
∀x Building(x) → ArchitecturalStructure(x)

(RDF) graph / facts / data

Tower(EiffelTower) Building(Shard)
City(Paris) City(London)
location(EiffelTower, Paris) height(EiffelTower, 324m)
location(Shard, London) height(Shard, 310m)
capital_of(Paris, France) capital_of(Shard, London)
capital_of(Shard, London) capital_of(London, UK)
membre_of(France, EU) member_of(UK, EU)
membre_of(UK, EU)
Knowledge Graph Semantics

∀x Tower(x) → ArchitecturalStructure(x)
∀x Building(x) → ArchitecturalStructure(x)

Tower(EiffelTower) Building(Shard)
City(Paris) City(London)
location(EiffelTower, Paris) height(EiffelTower, 324m)
location(Shard, London) height(Shard, 310m)
capital_of(Paris, France) capital_of(London, UK)
member_of(France, EU) member_of(UK, EU)
Knowledge Graph Rules

Knowledge base/graph

∀x Tower(x) → ArchitecturalStructure(x)
∀x Building(x) → ArchitecturalStructure(x)
∀x, y, z location(x, y) ∧ capital_of(y, z) → location(x, z)
∀x, y, z location(x, y) ∧ member_of(y, z) → location(x, z)

Tower(EiffelTower) Building(Shard)
City(Paris) City(London)
location(EiffelTower, Paris) height(EiffelTower, 324m)
location(Shard, London) height(Shard, 310m)
capital_of(Paris, France) capital_of(London, UK)
member_of(France, EU) member_of(UK, EU)
Knowledge Graph Rules

Knowledge base/graph

**Rules:**

- $\text{Tower}(x) \rightarrow \text{ArchitecturalStructure}(x)$
- $\text{Building}(x) \rightarrow \text{ArchitecturalStructure}(x)$
- $\text{location}(x, y) \land \text{capital_of}(y, z) \rightarrow \text{location}(x, z)$
- $\text{location}(x, y) \land \text{member_of}(y, z) \rightarrow \text{location}(x, z)$

**Example:**

- $\text{Tower}(\text{EiffelTower})$
- $\text{Building}(\text{Shard})$
- $\text{City}(\text{Paris})$
- $\text{City}(\text{London})$
- $\text{location}(\text{EiffelTower}, \text{Paris})$
- $\text{height}(\text{EiffelTower}, 324\text{m})$
- $\text{location}(\text{Shard}, \text{London})$
- $\text{height}(\text{Shard}, 310\text{m})$
- $\text{capital_of}(\text{Paris}, \text{France})$
- $\text{capital_of}(\text{London}, \text{UK})$
- $\text{member_of}(\text{France}, \text{EU})$
- $\text{member_of}(\text{UK}, \text{EU})$
Knowledge Graph Query Answering

Knowledge base/graph

Tower(x) \rightarrow ArchitecturalStructure(x)
Building(x) \rightarrow ArchitecturalStructure(x)
location(x, y) \land capital_{of}(y, z) \rightarrow location(x, z)
location(x, y) \land member_{of}(y, z) \rightarrow location(x, z)

Tower(EiffelTower) \quad Building(Shard)
City(Paris) \quad City(London)
location(EiffelTower, Paris) \quad height(EiffelTower, 324m)
location(Shard, London) \quad height(Shard, 310m)
capital_{of}(Paris, France) \quad capital_{of}(London, UK)
member_{of}(France, EU) \quad member_{of}(UK, EU)

\mathcal{K} \models ArchitecturalStructure(EiffelTower) \land location(EiffelTower, EU)
Knowledge Graph Query Answering

Knowledge base/graph

Tower($x$) $\rightarrow$ ArchitecturalStructure($x$)
Building($x$) $\rightarrow$ ArchitecturalStructure($x$)
location($x, y$) $\land$ capital_of($y, z$) $\rightarrow$ location($x, z$)
location($x, y$) $\land$ member_of($y, z$) $\rightarrow$ location($x, z$)

Tower(EiffelTower) Building(Shard)
City(Paris) City(London)
location(EiffelTower, Paris) height(EiffelTower, 324m)
location(Shard, London) height(Shard, 310m)
capital_of(Paris, France) capital_of(London, UK)
member_of(Paris, France) member_of(UK, EU)

$\mathcal{K} \models$ ArchitecturalStructure(EiffelTower) $\land$
location(EiffelTower, EU)
Views & Rules

- Integration & restructuring (e.g., introduce EUStruc)
- Security (e.g., only allow access to EUStruc)
- Simplification (e.g., use EUSruc in other queries/rules)
- Optimisation (e.g., materialize EUStruc)

Rules

- Recursive definitions (e.g., location)
- Critical for, e.g., part-whole, connectivity, causation, ...
Knowledge Graph Systems
• Materialization reasoning seems ideal for data-centric applications
  • Can support expressive ontology/rule languages
  • Fast query answering over very large graphs

• Challenges
  • Materialisation can be costly in time and memory
  • How to deal with (rapidly) changing data
  • Reliability and correctness!

• Solution: RDFox
  • Optimised materialization exploiting modern multi-core architectures
  • Incremental maintenance as data changes
  • Formally specified and proven-correct algorithms
• Novel algorithms developed at Oxford
  • Proven correctness & performance
• Novel algorithms developed at Oxford
  • Proven correctness & performance
• **Novel algorithms developed at Oxford**
  • Proven correctness & performance
• **Optimized in-memory data structures**
  • $>10^9$ triples on 128 Gb entry level server
  • $>10^{10}$ triples on 1 Tb server
• Novel algorithms developed at Oxford
  • Proven correctness & performance
• Optimized in-memory data structures
  • $>10^9$ triples on 128 Gb entry level server
  • $>10^{10}$ triples on 1 Tb server
• Parallelised materialisation
  • Dynamic distribution of workload
  • Mostly lock-free data structures
• Novel algorithms developed at Oxford
  • Proven correctness & performance
• Optimized in-memory data structures
  • $>10^9$ triples on 128 Gb entry level server
  • $>10^{10}$ triples on 1 Tb server
• Parallelised materialisation
  • Dynamic distribution of workload
  • Mostly lock-free data structures
• Incremental addition and retraction
  • Novel B/F materialisation maintenance algorithm
Oxford Semantic Technologies

High performance knowledge graph and semantic reasoning engine.

Free Trial
Extensions (beyond OWL RL)

- Arbitrary rules
  - No restriction to OWL RL (tree-shaped) rules

- Data types and values
  - Numbers, strings, dates, ...
  - Built in functions and aggregation

- Value invention
  - Add new (possibly computed) values to graph
  - Add new URI nodes to graph

- Constraints and negation as failure
  - SHACL+
System Architecture
Knowledge Graph Applications
Configuration management

- Components
- Their attributes & constraints
- Definitions of compatibility & valid configurations

\[ \text{FILTER} (\text{pv} = \text{rv} \land \text{pc} \geq \text{rc}) \]
Data Integration

• Integrate data from multiple sources
  • Companies
  • Executives
  • Stock markets
  • Geonames
  • Articles from WSJ, Factiva, ...

• Query integrated data
  • Competitor companies that are NASDAQ listed and have subsidiaries in same or related sector
  • Article published between 2020-05-24 and 2020-05-26 that talk about company C and mention an African country
Wrap-up
Summary

• **KGs are powerful tool** for representing & reasoning about knowledge

• **Many applications**: configuration, data integration, fraud detection, ...

• **Technical challenges**: scalability, correctness, **knowledge engineering** ...

• **Solutions** based on **foundational research** + **systems engineering**
Thanks for Listening
Any Questions?

Background reading:


