2. Theoretical Foundations

Olaf Hartig
University of Waterloo
“Ingredients” for LD Query Execution

- Data retrieval approach
  - Data source selection
  - Data source ranking (optional, for optimization)

- Result construction approach
  - i.e., query-local data processing

- Combining data retrieval and result construction

---

<table>
<thead>
<tr>
<th>?actor</th>
<th>?loc</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://mdb.../Ric">http://mdb.../Ric</a></td>
<td><a href="http://geo.../Rome">http://geo.../Rome</a></td>
</tr>
</tbody>
</table>
Outline

- Basics of the SPARQL Query Language
- Data Model for Linked Data Queries
- Full-Web Query Semantics
  - Definition
  - Theoretical Properties
- Reachability-Based Query Semantics
  - Definition
  - Theoretical Properties
General Idea of SPARQL

- **SPARQL**: Declarative query language for RDF data

- **Main idea**: pattern matching
  - Query describes a pattern to be found in RDF data
  - Basically, query patterns consist of RDF triples with variables

**Basic Query Pattern:**

\[
( ?p , \text{affiliated with} , \text{orgaX} ) ,
( ?p , \text{interested in} , ?i )
\]
General Idea of SPARQL

- **SPARQL**: Declarative query language for RDF data
- **Main idea**: pattern matching
  - Query describes a pattern to be found in RDF data
  - Basically, query patterns consist of RDF triples with variables
  - Any part of the queried data that matches the query pattern contributes a solution to the query result

**Basic Query Pattern:**
( ?p , affiliated with , orgaX ) ,
( ?p , interested in , ?i )

**Data:**
( alice , affiliated with , orgaX ) ,
( bob , affiliated with , acme ) ,
( alice , interested in , yoga ) ,
( alice , interested in , tea ) ,
( bob , interested in , tea ) ,

**Query Result:**

<table>
<thead>
<tr>
<th>?p</th>
<th>?i</th>
</tr>
</thead>
<tbody>
<tr>
<td>alice</td>
<td>yoga</td>
</tr>
<tr>
<td>alice</td>
<td>tea</td>
</tr>
</tbody>
</table>
Some Terminology

- Any query result is represented by a set of valuations
  \[ \Omega = \{ \mu_1, \ldots, \mu_n \} \]

- Valuation: mapping of query variables to RDF terms
  \[ \mu = \{ ?p \rightarrow \text{alice}, ?i \rightarrow \text{yoga} \} \]

- Any valuation in a query result is a solution

- Two valuations \( \mu_1 \) and \( \mu_2 \) are compatible if they agree on the binding for any shared variable
  - i.e., \( \mu_1(\text{?v}) = \mu_2(\text{?v}) \) for all \( \text{?v} \in \text{dom}(\mu_1) \cap \text{dom}(\mu_2) \)
Query Patterns and their Semantics

• **Triple pattern**: An RDF triple that may have a variable as subject, predicate, or object, respectively
  
  • \( \text{eval}(tp, G) := \{ \mu \mid \mu[tp] \in G \text{ and } \text{dom}(\mu) = \text{vars}(tp) \} \)

• **Group graph pattern**: Conjunction of two query patterns
  
  • \( \text{eval}(P_1 \text{ AND } P_2, G) := \{ \mu_1 \cup \mu_2 \mid \mu_1 \in \text{eval}(P_1, G) \text{ and } \mu_2 \in \text{eval}(P_2, G) \text{ and } \mu_1 \text{ and } \mu_2 \text{ are compatible} \} \)

• Associative and commutative

• **Basic graph pattern (BGP)**: A set of triple patterns
  
  • \( B = \{ tp_1, \ldots, tp_n \} \) is equivalent to \( (tp_1 \text{ AND } \ldots \text{ AND } tp_n) \)

• Filter, union, optional, etc.
Outline

- Basics of the SPARQL Query Language ✓
- Data Model for Linked Data Queries
- Full-Web Query Semantics
  - Definition
  - Theoretical Properties
- Reachability-Based Query Semantics
  - Definition
  - Theoretical Properties
Data Model

- Static view (i.e., no changes during computation)
- Web of Linked Data: \( W = ( D, \text{adoc}, data ) \)
  - \( D \) – set of symbols that represent Web documents from which Linked Data can be extracted
  - \( \text{adoc} \) – partial mapping from URIs to \( D \)
  - \( data \) – total mapping from \( D \) to finite sets of RDF triples

\[
D = \left\{ \begin{array}{l}
\end{array} \right. \quad \text{adoc} = \left\{ \begin{array}{l}
\end{array} \right. \quad \text{data} = \left\{ \begin{array}{l}
\end{array} \right.
\]

- All data in \( W \): \( \text{AllData}(W) := \bigcup_{d \in D} data(d) \)
Linked Data Query

\[ Q(W) \]
SPARQL-Based Linked Data Query

\[ Q^P(W) = \{ \mu_1, \mu_2, \ldots \} \]

( ?p , affiliated with , orgaX ) ,
( ?p , interested in , ?i )
Outline

- Basics of the SPARQL Query Language ✓
- Data Model for Linked Data Queries ✓
- Full-Web Query Semantics
  - Definition
  - Theoretical Properties
- Reachability-Based Query Semantics
  - Definition
  - Theoretical Properties
$Q^P(W) := \text{eval}(P, \text{AllData}(W))$
Computability

\[ Q^P(W) := \text{eval}(P, \text{AllData}(W)) \]

- (Ordinary) Turing machines are unsuitable:
  - Limited data access capabilities not properly captured

- Web machines
  - Abiteboul and Vianu, 1997
  - Mendelzon and Milo, 1997
LD Machine

- Multi-tape Turing machine

  ➔ Input
  ➔ Work
  ➔ Output
LD Machine

- Multi-tape Turing machine
  - Web Input: $\# \text{enc}(u_1) \text{enc}(adoc(u_1)) \# \text{enc}(u_2) \text{enc}(adoc(u_2)) \# \cdots$
  - Input
  - Work
  - Output
LD Machine

- **Multi-tape Turing machine**
  - Web Input
  - Input
  - Work
  - Output

- **Access to Web Input is restricted**
  - Only by performing a particular procedure in a particular state
Finitely Computable LD Queries

→ Web Input

→ Input

→ Work

→ Output

• For \( Q \) exists an LD machine \( M_Q \) such that for any \( W \) holds:
  • \( M_Q \) halts after a finite number of computation steps, and
  • \( M_Q \) outputs the complete result \( Q(W) \)
Eventually Computable LD Queries

- Web Input
  - # enc(u₁) enc(adoc(u₁)) # enc(u₂) enc(adoc(u₂)) # · · ·
- Input
- Work
- Output
  - # enc(μ₁) # enc(μ₂)

- For Q exists an LD machine $M_Q$ such that for any $W$ holds:
  1. Output always encodes a subset of query result $Q(W)$, and
  2. Each $μ ∈ Q(W)$ eventually appears on the output

✓ No guarantee for termination
For \( Q \) exists an LD machine \( M_Q \) such that for any \( W \) holds:

1. Output always encodes a subset of query result \( Q(W) \), and
2. Each \( \mu \in Q(W) \) eventually appears on the output

✗ No guarantee for termination
Theorem [Har12]: If a satisfiable SPARQL$_{LD}$ query $Q^P$ under full-Web semantics is monotonic, then it is eventually computable (but not finitely computable); otherwise, it is not even eventually computable.

- Main reason: Set of all URIs is infinite (but countable)
Satisfiability and Monotonicity

Satisfiability:
\[ Q(\mathcal{R}) = \{ \ldots, r_i, \ldots \} \neq \emptyset \]

Monotonicity:
\[ Q(\mathcal{R}) \subseteq Q(\mathcal{T}) \]

Unfortunately:
Satisfiability and monotonicity of SPARQL query patterns is undecidable

Proposition [Har12]: Let \( Q^P \) be a SPARQL\(_{LD} \) query under full-Web semantics.

- \( Q^P \) is satisfiable \( \iff \) its query pattern \( P \) is satisfiable
- \( Q^P \) is monotonic \( \iff \) its query pattern \( P \) is monotonic
Computability (Full-Web Semantics)

\[ Q^P(W) := \text{eval}(P, \text{AllData}(W)) \]

**Theorem [Har12]:** If a satisfiable SPARQL$_{LD}$ query \( Q^P \) under full-Web semantics is *monotonic*, then it is eventually computable (but not finitely computable); otherwise, it is not even eventually computable.

- **Main reason:** Set of all URIs is infinite (but countable)
Outline

- Basics of the SPARQL Query Language  ✔
- Data Model for Linked Data Queries  ✔
- Full-Web Query Semantics
  - Definition
  - Theoretical Properties  ✔
- Reachability-Based Query Semantics
  - Definition
  - Theoretical Properties
Reachability-Based Query Semantics
Reachability-Based Query Semantics

- Seed URIs $\mathbf{S} \subseteq \mathbf{U}$
Reachability-Based Query Semantics

- **Reachability criterion** $c$
  - (Turing) computable function $c: T \times U \times P \rightarrow \{\text{true, false}\}$

- **Seed URIs** $S \subset U$
Reachability-Based Query Semantics

\[ Q^{P,S}_{c}(W) := \text{eval}(P, \text{AllData}(W^*)) \]
(Reachability-Based) $c_{\text{All}}$-Semantics

$$Q_{c_{\text{All}}}^{P,S}(W) := \text{eval}(P, \text{AllData}(W^*))$$
(Reachability-Based) $c_{\text{None}}$-Semantics

$$Q_{c_{\text{None}}}^{P,S}(W) := \text{eval}(P, \text{AllData}(W^*))$$
$(\text{Reachability-Based})\ c_{\text{Match}}\text{-Semantics}$

$Q_{c_{\text{Match}}}^{P,S}(W) := \text{eval}(P,\text{AllData}(W^*))$
Satisfiability and Monotonicity

Satisfiability:
\[ Q(\ldots, r_i, \ldots) \neq \emptyset \]

Monotonicity:
\[ Q(\ldots) \subseteq Q(\ldots) \]

Proposition [Har12]: Let \( c \) be a reachability criterion and \( Q^P_c S \) be a SPARQL\(_{LD} \) query under \( c \)-semantics.
- \( Q^P_c S \) is satisfiable \( \iff \) its SPARQL expression \( P \) is satisfiable
- \( Q^P_c S \) is monotonic \( \Rightarrow \) its SPARQL expression \( P \) is monotonic

Counterexample: Let \( Q^P_c S \) be a SPARQL\(_{LD} \) query under \( c \)-semantics. If \( c = c_{\text{None}} \) and \( |S| = 1 \), then \( Q^P_c S \) is monotonic.
LD Machine-Based Computability

- If we restrict our data model such that Webs of Linked Data are finite (i.e., consist of a finite number of documents), then:

**Theorem [Har12]:** For any reachability criterion $c$, any SPARQL based Linked Data query under $c$-semantics is finitely computable.

For results w.r.t. the unrestricted data model (i.e., Webs of Linked Data that may be infinitely large), refer to [Har12].
Outline

- Basics of the SPARQL Query Language ✓
- Data Model for Linked Data Queries ✓
- Full-Web Query Semantics
  - Definition
  - Theoretical Properties ✓
- Reachability-Based Query Semantics
  - Definition
  - Theoretical Properties ✓

Next part: 3. Source Selection Strategies ...