Data Integration: RDF

Jan Chomicki

University at Buffalo and Warsaw University

May 17, 2007

SPARQL

A query language for RDF based on graph matching.

Parts of queries

1. pattern matching:
   - optional parts
   - unions of patterns
   - nesting
   - filtering
2. solution modifiers:
   - projection
   - distinct
   - order,...
3. output:
   - yes/no queries
   - variable values
   - new triples

Basic notions

Domains

- I: URIs
- B: blank nodes
- L: literals
- $IL = I \cup L$
- $T = I \cup B \cup L$
- V: variables

RDF

Triple: $(s, p, o) \in (I \cup B) \times I \times T$.
Graph: set of triples.

Graph patterns

- basic: tuples in $(IL \cup V) \times (I \cup V) \times (IL \cup V)$
- binary: $(P_1$ AND $P_2)$, $(P_1$ OPT $P_2)$, $(P_1$ UNION $P_2)$
- unary: $(P$ FILTER $R)$ where R is a built-in condition (a Boolean combination of equality and boundness tests)

Semantics

Mappings

- partial function from V to T
- two mappings are compatible if they agree on the common part of the domains.

Operations on sets of mappings

$\Omega_1 \bowtie \Omega_2 = \{\mu_1 \cup \mu_2 \mid \mu_1 \in \Omega_1, \mu_2 \in \Omega_2$ compatible mappings$\}$
$\Omega_1 \cup \Omega_2 = \{\mu \mid \mu \in \Omega_1$ or $\mu \in \Omega_2\}$
$\Omega_1 \setminus \Omega_2 = \{\mu \in \Omega_1 \mid$ for all $\mu' \in \Omega_1,\ \mu$ and μ' not compatible$\}$
$\Omega_1 \leftarrow \Omega_2 = (\Omega_1 \bowtie \Omega_2) \cup (\Omega_1 \setminus \Omega_2)$ (left outerjoin).

Pattern evaluation over an RDF graph D

1. $[\![t]\!]_D = \{\mu \mid dom(\mu) = var(t), \mu(t) \in D\}$
2. $[\![(P_1$ AND $P_2)]\!]_D = [\![P_1]\!]_D \bowtie [\![P_2]\!]_D$
3. $[\![(P_1$ OPT $P_2)]\!]_D = [\![P_1]\!]_D \leftarrow [\![P_2]\!]_D$
4. $[\![(P_1$ UNION $P_2)]\!]_D = [\![P_1]\!]_D \cup [\![P_2]\!]_D$
5. $[\![(P$ FILTER $R)]\!]_D = \{\mu \in [\![P]\!]_D \mid \mu$ satisfies $R\}$

Theoretical properties of graph patterns [PAG06]

Algebraic properties

1. AND and UNION are associative and commutative
2. AND distributes over UNION
3. OPT distributes over UNION (from both left and right)
4. FILTER distributes over UNION

Normal form

Each graph pattern is equivalent to a UNION of UNION-free expressions.

The pattern evaluation problem

INPUT: RDF graph D, graph pattern P, mapping μ.
OUTPUT: $\mu \in [\![P]\!]_D$?

Complexity of pattern evaluation

- in PTIME ($O(|P| \cdot |D|)$) for AND-FILTER patterns
- NP-complete for AND-FILTER-UNION patterns
- PSPACE-complete for AND-FILTER-UNION-OPT patterns

Well-designed patterns

Definition

A UNION-free pattern P is well designed if for every sub-pattern $P' = (P_1 \text{ OPT } P_2)$ of P and every variable $?X$ occurring in P:

if $?X$ occurs both inside P_2 and outside P', then it also occurs in P_1.

Properties of well-designed patterns

- for well-designed patterns the algebraic and the normative operational semantics coincide
- OPT associates with AND
- every well-designed pattern is equivalent to one in the form

$$(\cdots((t_1 \text{ AND } \cdots \text{ AND } t_k) \text{ OPT } O_1)\cdots \text{ OPT } O_n)$$

where each t_i is a triple pattern and each O_j is in the above form.


J. Pérez, M. Arenas, and C. Gutierrez.
Semantics and Complexity of SPARQL.
In *International Semantic Web Conference*, pages 30–43. Springer, LNCS 4273, 2006.