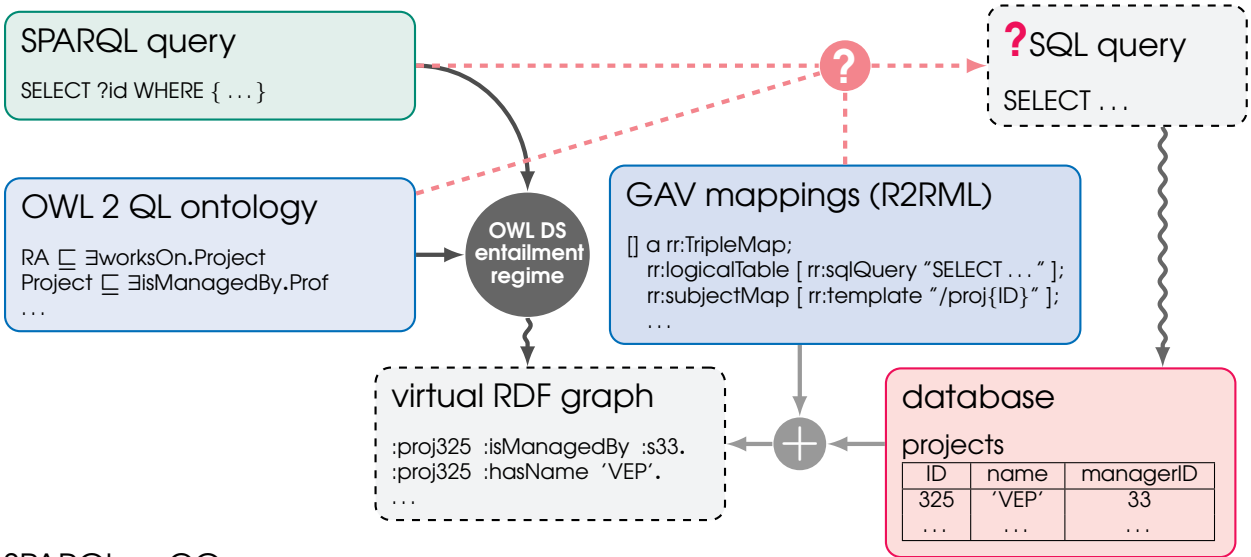
Answering SPARQL Queries over Databases under OWL 2 QL Entailment Regime

Roman Kontchakov, Martin Rezk,
Mariano Rodríguez-Muro, Guohui Xiao
and Michael Zakharyaschev

Birkbeck, University of London, Free University of Bozen-Bolzano, IBM

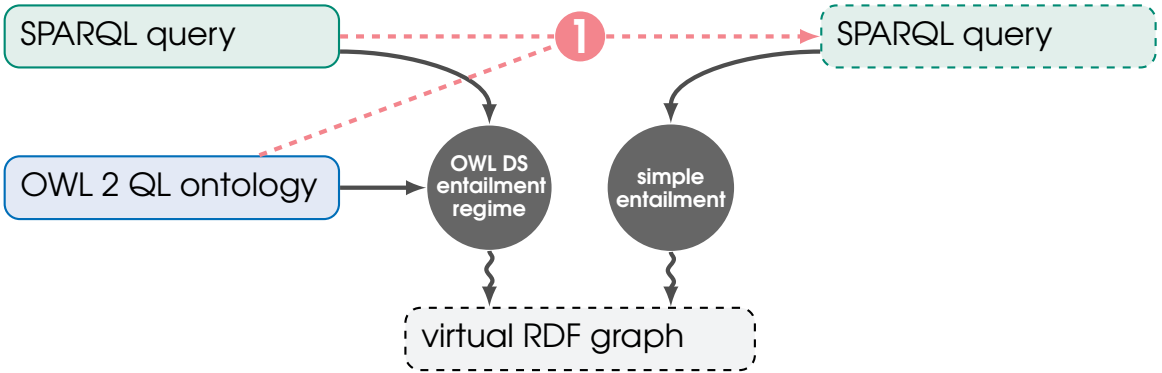
Ontology-Based Data Access



SPARQL vs CQs

- **+** meta-variables (class and property variables)
- **+** OPTIONAL, etc.
- **-** variables (and blank nodes) are mapped onto RDF graph (not the labelled nulls)

Step 1. OWL 2 QL Entailment Regime



1.1 use BIND:

```
SELECT ?x ?C WHERE { ?x rdf:type ?C. ?C rdfs:subClassOf ub:Student. }
```

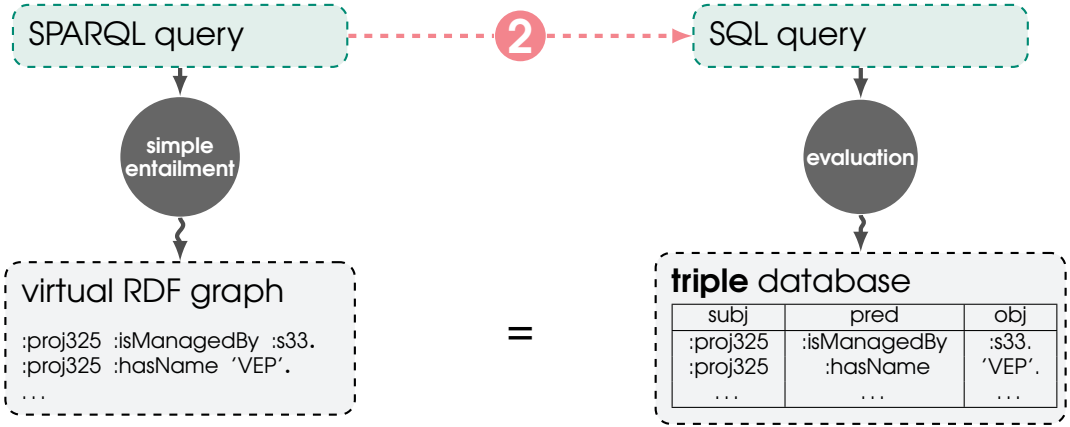
↓

```
SELECT ?x ?C WHERE { { ?x rdf:type ?C. BIND (ub:Student AS ?C). } UNION
                     { ?x rdf:type ?C. BIND (ub:GradStudent AS ?C). } UNION
                     { ?x rdf:type ?C. BIND (ub:UGStudent AS ?C). } }
```

1.2 use tree-witness rewriting for **ClassAssertion**($\boldsymbol{C}$, $\boldsymbol{I}$)

$\boldsymbol{C}$ is a **complex** OWL 2 QL concept (with $\sqcap$ and $\exists \boldsymbol{R}.\boldsymbol{C}$)

Step 2: From SPARQL to SQL (1)



$s\ p\ o \xrightarrow{\tau}$

$\pi_s \varrho_{s/\text{subj}} \sigma_{\text{pred}=p} \sigma_{\text{obj}=o}$ *triple*, if $s \in \mathbf{V}$ and $p, o \in \mathbf{I} \cup \mathbf{L}$

$\pi_{s,o} \varrho_{s/\text{subj}} \varrho_{o/\text{obj}} \sigma_{\text{pred}=p}$ *triple*, if $s, o \in \mathbf{V}, s \neq o$, and $p \in \mathbf{I} \cup \mathbf{L}$

$\pi_s \varrho_{s/\text{subj}} \sigma_{\text{subj}=\text{obj}} \sigma_{\text{pred}=p}$ *triple*, if $s, o \in \mathbf{V}, s = o$, and $p \in \mathbf{I} \cup \mathbf{L}$

...

Step 2: From SPARQL to SQL (2)

- $\text{UNION}(P_1, P_2) \xrightarrow{\tau} \mu_{U_2 \setminus U_1}\tau(P_1) \cup \mu_{U_1 \setminus U_2}\tau(P_2)$

μ_U padding with *null* for U

P_1

?p	?t
:j	079

P_2

?p	?e
:j	h@a.com

$\text{UNION}(P_1, P_2)$

?p	?t	?e
:j	079	
:j		h@a.com

p	t	e
:j	079	*null*
:j	*null*	h@a.com

- $\text{FILTER}(P, F) \xrightarrow{\tau} \sigma_{\tau(F)}\tau(P)$

use **three-valued logic** (with $\top$, $\bot$ and ε): $\tau(\mathit{bound}(?x)) = \neg \mathit{isNull}(x)$

- $\text{JOIN}(P_1, P_2) \xrightarrow{\tau} \bigcup_{\substack{V_1, V_2 \subseteq U_1 \cap U_2 \\ V_1 \cap V_2 = \emptyset}} \mu_{V_1 \cup V_2}\left[(\pi_{U_1 \setminus V_1}\sigma_{\mathit{isNull}(V_1)}\tau(P_1)) \bowtie (\pi_{U_2 \setminus V_2}\sigma_{\mathit{isNull}(V_2)}\tau(P_2))\right]$

Step 2: From SPARQL to SQL (3)

- if each variable by shared P_1 and P_2 is necessarily bound then

$$\tau(\text{JOIN}(P_1, P_2)) = \tau(P_1) \bowtie \tau(P_2)$$

$$\tau(\text{OPT}(P_1, P_2, F)) = \tau(P_1) ⟕_{\tau(F)} \tau(P_2)$$

LEFT JOIN in SQL with the filter in ON

$\nu(P)$ — variables that are not necessarily bound in solution mappings for P:

$\nu(BGP) = \emptyset$

$\nu(\text{UNION}(P_1, P_2)) = (var(P_1) \setminus var(P_2)) \cup \nu(P_1) \cup (var(P_2) \setminus var(P_1)) \cup \nu(P_2)$

$\nu(\text{JOIN}(P_1, P_2)) = \nu(P_1) \cup \nu(P_2)$

...

NB: $\text{OPT}(P_1, P_2, F) = \text{FILTER}(\text{JOIN}(P_1, P_2), F) \cup \underbrace{\text{DIFF}(P_1, P_2, F)}$

$\{s_1 \in P_1 \mid$ for all $s_2 \in P_2$, either $s_1 \not\sim s_2$ or $F^{s_1 \oplus s_2} \neq \top\}$

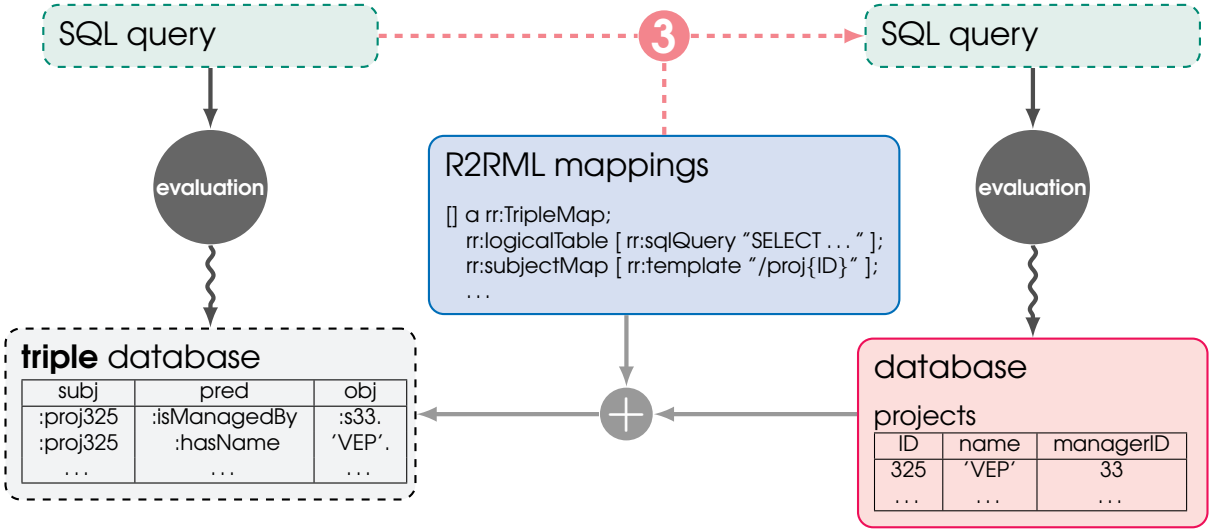
~~$= \bot$~~ 'has an effective boolean value of false'

NB: $\text{DIFF}(P_1, P_2, F) \neq$

$\{s_1 \in P_1 \mid s_1 \not\sim s_2$, for all $s_2 \in P_2\} \cup$

$\{s_1 \in P_1 \mid$ there is $s_2 \in P_2$ with $s_1 \sim s_2$ and $F^{s_1 \oplus s_2} = \bot\}$

Step 3: From Triples to Relations



Semantic Query Optimisation:
using **primary** and **foreign key** constraints to remove redundant joins

T-mappings: compile the class and property **hierarchies** into the mappings
(and use SQO to reduce them)

remove subqueries with incompatible **IRI templates** and de-IRI joins

Implementation:

ontop framework

query execution time in ms

query	LUBM-1				LUBM-9			LUBM-100		LUBM-200	LUBM-500
	Ontop	OWL BGP/H	OWL BGP/P	Pellet	Ontop	OWL BGP/H	Pellet	Ontop	Pellet	Ontop	Ontop
q_1	2	8	29	1	3	97	1	3	1	3	2
q_2	2	25	11 137	19	3	2 531	256	16	30 593	36	88
q_3	1	6	86	9	2	78	158	2	2 087	63	12
q_4	13	7	19	14	15	44	164	27	2 093	24	22
q_5	16	12	4 451	10	22	98	158	32	2 182	28	23
q_6	455	27	32	21	5 076	411	317	58 968	10 781	123 578	434 349
q_7	5	21	34 005	10	6	429	157	8	2 171	8	9
q_8	726	195	95 875	80	760	917	192	796	2 131	820	855
q_9	60	972	168 978	78	668	189 126	857	7 466	12 125	15 227	44 598
q_{10}	2	6	126	9	3	97	158	2	2 134	3	2
q_{14}	91	20	24	15	1 168	329	287	13 524	4 457	29 512	92 376
q_4'	93	58	190	46	99	98	767	92	4 422	95	107
q_4''	108	21	35	63	122	72	719	115	9 179	108	127
q_9'	257	716	91 855	174	4 686	40 575	1 385	54 092	19 945	115 110	295 228
q_9''	557	951	65 916	102	6 093	178 401	1 214	67 123	19 705	151 376	356 176
q_2^{obg}	150	30	57 141	29	9 992	520	348	39 477	5 411	79 351	206 061
q_4^{obg}	6	7	241	25	31	40	273	7	3 969	7	494
q_{10}^{obg}	641	760	31 269	253	6 998	149 191	2 258	163 308	17 929	174 362	459 669

- **SPARQL** over databases with **OWL2QL** inferencing and **R2RML** mappings
- JOIN/OPT optimisations, SQO, T-mappings, IRI templates handling
- **open-source** software (Apache 2.0 license) `http://ontop.inf.unibz.it`

support for DB2, Oracle, MS SQL Server, MySQL, Postgres, HSQL, H2