Towards Equivalences for Federated SPARQL Queries

Carlos Buil Aranda (1) and Axel Polleres (2)

(1) Department of Computer Science, Pontificia Universidad Católica, Chile cbuil@ing.puc.cl
(2) Vienna University of Economy and Business (WU) axel.polleres@wu.ac.at
What usually happens when we send one SPARQL (Federated) query to an endpoint?
Some features are not supported
Partial Results
Timeouts
Consider this federated SPARQL query:

```sparql
select * where {
  service<http://mgi.bio2rdf.org/sparql> {  
    # from MGI to HGCN: low selectivity from MGI
  }
  # low selectivity from HGCN
  service<http://hgnc.bio2rdf.prg/sparql> {   
  }
}
```
Consider this federated SPARQL query:
What can we do?

Find alternatives to the VALUES operator...
Some possible alternatives:

We can use the SPARQL FILTER operator:

$$P_{2}^{\text{FILTER}_{P_1}} = \{ P_2 \text{ FILTER } \bigvee_{\mu \in [P_1]_G} \bigwedge_{v \in \text{dom}(\mu) \cap \text{vars}(P_2)} v = \mu(v) \}$$

Or use the SPARQL UNION operator:

$$P_{2}^{\text{UNION}_{P_1}} = \{(\mu_1^{\text{FILTER}}(P_2)) \text{ UNION } \ldots \text{ UNION } (\mu_n^{\text{FILTER}}(P_2))\}$$

Where:

$$\mu^{\text{FILTER}}(P) = P \text{ FILTER } (\bigwedge_{v \in \text{dom}(\mu) \cap \text{vars}(P_2)} v = \mu(v))$$
However, consider this query now:

Assume:

\[ P_1 = (\text{?X}, \text{b}, \text{c}), \text{c = I} \text{ and } P_2 = (\text{?Y,d,e}) \text{ UNION (?X,d,e ))} \]

With Data:

\[ \text{Local default graph } G_1 = \{(a, b, c)\} \]

\[ \text{Remote service’ default graph } G_2 = \{(a, d, e)\} \]
We obtain: $[P_1]_{G_1} = \{[?X \rightarrow a]\}$ and $[P_2]_{G_2} = \{[?X \rightarrow a], [?Y \rightarrow a]\}$

The second solution for $[P_2]_{G_2}$, i.e. $\mu_2 = [?Y \rightarrow a]$ is compatible with the single solution for $[P_1]_{G_1}$, i.e., $\mu_1 = [?Y \rightarrow a]$ Yielding overall $\mu = [?X \rightarrow a, ?Y \rightarrow a]$.

However, $P_2^{\text{FILTER}_{P_1}} = P_2^{\text{UNION}_{P_1}} = \{(?Y,d,e) \text{ UNION } (?X,d,e)\}$ $\text{FILTER}( ?X = a)$ which would not yield $\mu$ as a solution.
Let’s try to fix this:

We have a look at the Strongly Bound variables from [BAC2011]

- if $P = t$, where $t$ is a triple pattern, then $SB(P) = \text{var}(t)$;
- if $P = (P_1 \text{ AND } P_2)$, then $SB(P) = SB(P_1) \cup SB(P_2)$;
- if $P = (P_1 \text{ UNION } P_2)$, then $SB(P) = SB(P_1) \cap SB(P_2)$;
- if $P = (P_1 \text{ OPT } P_2)$ or $P = (P_1 \text{ FILTER } R)$, then $SB(P) = SB(P_1)$;
- if $P = (P_1 \text{ FILTER } R)$, then $SB(P) = SB(P_1)$;
- if $P = (\text{SERVICE } c \ P_1)$, with $c \in I$, or $P = (\text{SERVICE } ?X \ P_1)$, with $?X \in V$, then $SB(P) = \emptyset$;
- if $P = (P_1 \text{ VALUES } S \{A_1, \ldots, A_n\})$, then $SB(P) = SB(P_1) \cup \{?X \mid ?X \text{ is in } S \text{ and for every } i \in \{1, \ldots, n\}, \text{ it holds that } ?X \in dom(\mu_{S,A_i})\}$.
- if $P = (\text{SELECT } W \ P_1)$, then $SB(P) = (W \cap SB(P_1))$. 
Using the previous Strongly Bound Definition:

**Lemma 1.** Given a SPARQL pattern $P$ with $v \in SB(P)$, let $\mu_e = [v \to e]$ for an $e \in I \cup L$, then $[\mu_e(P)]_G \otimes \mu_e = \{ \mu \in [P]_G | v \in \text{dom}(\mu) \land \mu(v) = e \}$.

i.e. we won’t get additional results when replacing SB variables

What about blank nodes combined with unbound variables? they are not shared between endpoints

$$\mu^B(P_2) = \{ \mu(P_2) \text{ FILTER}(\neg(\forall v \in \text{dom}(\mu) \cap \text{vars}(P_2) \text{ isBlank}(\mu(v)))) \}$$
Two theorems to fix the problems (1)

Theorem 1. Let $P = P_1 \text{ AND (SERVICE } c \text{ P}_2\text{) such that } (\text{vars}(P_2) \cap \text{vars}(P_1)) \subseteq \text{SB}(P_2)$, i.e. all variables that participate in a join are strongly bound in the pattern appearing on the service side, and let $G_c$ be the default graph of service $c$ and let $P^\text{UNION}_{P_1}$ and $P^\text{FILTER}_{P_1}$ be as defined above, then

(i) $[P]_G = \bigcup_{\mu \in [P_1]_G} (\mu \Join [\mu^B(P_2)]_{G_c})$

(ii) $[P]_G = [P_1]_G \Join [P^\text{UNION}_{P_1}]_{G_c} = [P_1]_G \Join [P^\text{FILTER}_{P_1}]_{G_c}$
Two theorems to fix the problems (2)

We add $v = \mu(v) \lor \neg \text{bound}(v)$

to the FILTER and UNION operations

**Theorem 2.** Let $P = P_1 \text{ AND } (\text{SERVICE } c \ P_2)$ and $G_c$ be the default graph of service $c$ then

$$[P]_G = [P_1]_G \Join [P_2]_{\text{FILTER}^{P_1}_c} G_c = [P_1]_G \Join [P_2]_{\text{UNION}^{P_1}_c} G_c$$
Conclusions

• SPARQL Federated query still needs lots of work

• Some implementations contain the errors presented (Jena ARQ uses a nested loop join)

• Two theorems for fixing some problems
  • We can skip result limits from servers as well!

• Evaluation with real endpoints to be done

• Federated query evaluation strategies needed depending on the endpoint