# UPDATING RDFS ABOXES AND TBOXES IN SPARQL

Albin Ahmeti, Diego Calvanese and Axel Polleres

TU WIEN · UNIVERSITAS STUDIORUM BAUZANENSIS · WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

## Motivation

```
DELETE { ?X a :Child . }
INSERT { ?Y a :Mother . }
WHERE { ?X :hasParent ?Y . }
```

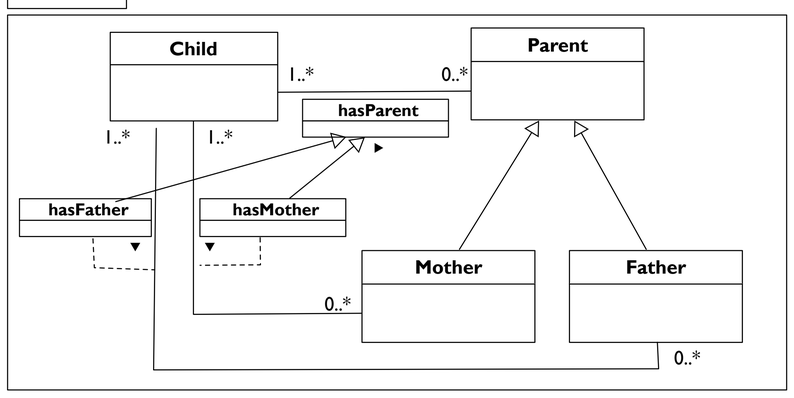
What does it mean to…

- DELETE an implicit triple?
- INSERT an already implied triple?
- WHERE matching variables on implicit triples?

How to preserve a materialized/reduced store?

**G**

**TBOX**




**ABOX**

| S | P | O |
|---|---|---|
| :joe | :hasMother | :jane |
| :joe | :hasParent | :jane |

*materialize(G)*

**ABOX-materialized**

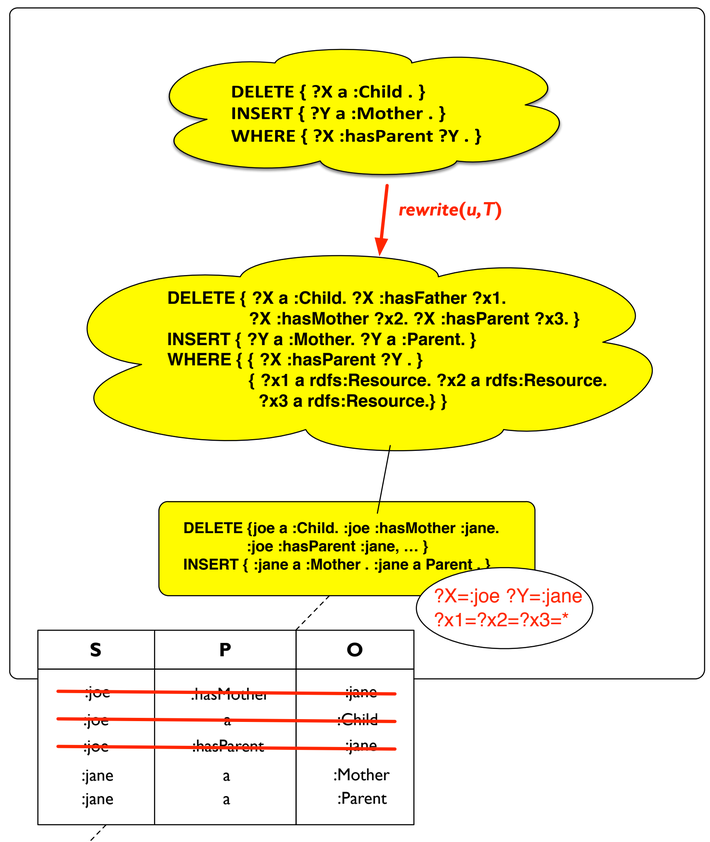
| S | P | O |
|---|---|---|
| :joe | :hasMother | :jane |
| :joe | a | :Child |
| :joe | :hasParent | :jane |
| :jane | a | :Mother |
| :jane | a | :Parent |

*reduce(G)*

**ABOX-reduced**

| S | P | O |
|---|---|---|
| :joe | :hasMother | :jane |

Update semantics Example (def. below): $Sem_2^{mat}$




**materialize(G):** *Minimal RDFS rules for ABox & TBox materialization*

$$\frac{?C \text{ sc } ?D. \quad ?S \text{ a } ?C.}{?S \text{ a } ?D.} \qquad \frac{?P \text{ dom } ?C. \quad ?S\ ?P\ ?O.}{?S \text{ a } ?C.} \qquad \frac{?C \text{ sc } ?D. \quad ?D \text{ sc } ?E.}{?C \text{ sc } ?E.}$$

$$\frac{?P \text{ sp } ?Q. \quad ?S\ ?P\ ?O.}{?S\ ?Q\ ?O.} \qquad \frac{?P \text{ rng } ?C. \quad ?S\ ?P\ ?O.}{?O \text{ a } ?C.} \qquad \frac{?P \text{ sp } ?Q. \quad ?Q \text{ sp } ?R.}{?P \text{ sp } ?R.}$$

**reduce(G):** *Update for ABox reduce*

```
DELETE {?S1 a ?D1. ?S2 a ?C2. ?S3 ?Q3 ?O3. ?O4 a ?C4.}
WHERE {{ ?C1 sc+ ?D1. ?S1 a ?C1. }
    UNION { ?P2 dom/sc* ?C2. ?S2 ?P2 ?O2. }
    UNION { ?P3 sp+ ?Q3. ?S3 ?P3 ?O3. }
    UNION { ?P4 rng/sc* ?C4. ?S4 ?P4 ?O4. }}
```

## State of art

What do off-the-shelf triple stores do?

- **Entailment** typically only handled at (bulk) loading *by **materialization*** but **not in the context of Updates**.
- No “standard” behavior for **Delete** & **Insert** upon materialized stores.
- Interplay of Entailments and Update left out in the SPARQL 1.1 spec.
- TBox **deletions** are ambiguous (*minimal cuts*)

## Contribution

- Discuss possible update semantics in the context of **materialized** and **reduced** stores & **RDFS**.
- Even in this restricted setting (RDFS) this turns out to be challenging

**Our setting (ABox):**

- Use pure RDFS as TBox language (without axiomatic triples, blank nodes)
- TBox is fixed
- Restrict on BGPs to only allow ABox Insert/Deletes

```
INSERT {:joe :hasFather ?Y }
 WHERE
{:joe :hasParent ?Y }
```

~~INSERT {:joe ?Y :foo}~~
~~ WHERE~~
~~{:joe rdf:type ?Y }~~

**Our setting (TBox):**

- Extend BGP to allow TBox updates
- Use the two rules for TBox materialization

```
DELETE {:A rdfs:subClassOf ?C . }
```

## Mat-preserving semantics: ABox updates

- $Sem_0^{mat}$ … baseline semantics

$$G^{\mathbf{Sem}_0^{mat}}_{u(P_d,P_i,P_w)} = materialize(G_{u(P_d,P_i,P_w)})$$

- $Sem_{1a}^{mat}$ … inspired by DRed: delete incl. effects and rederive upon inserts

$$G^{Sem_{1a}^{mat}}_{u(P_d,P_i,P_w)} = materialize(\mathcal{T} \cup (\mathcal{A} \setminus materialize(\mathcal{T} \cup \mathcal{A}_d)) \cup \mathcal{A}_i)$$

$$\mathcal{A}_d = \bigcup_{\theta \in ans(P_w,G)} gr(P_d\theta)$$

$$\mathcal{A}_i = \bigcup_{\theta \in ans(P_w,G)} gr(P_i\theta)$$

- $Sem_{1b}^{mat}$ … variant of DRed with two distinguished stores

$$G^{Sem_{1b}^{mat}}_{u(P_d,P_i,P_w)} = \mathcal{T} \cup \mathcal{A}'_{expl} \cup \mathcal{A}'_{impl}$$

$$\mathcal{A}'_{expl} = (\mathcal{A}_{expl} \setminus \mathcal{A}_d) \cup \mathcal{A}_i$$

$$\mathcal{A}'_{impl} = materialize(\mathcal{A}'_{expl} \cup \mathcal{T}) \setminus \mathcal{T}$$

- $Sem_2^{mat}$ … delete incl. causes and rewrite upon inserts

$$G^{Sem_2^{mat}}_{u(P_d,P_i,P_w)} = G_{u(P_d^{caus},P_i^{eff},\{P_w\}\{P_d^{fvars}\})}$$

$$P_d^{fvars} = \{\texttt{?x a rdfs:Resource.} \mid \text{for each } \texttt{?x} \in Var(P_d^{caus_{P_d}}) \setminus Var(P_d)\}$$

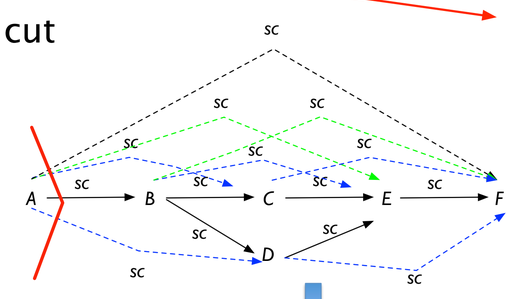
## Mat-preserving semantics: TBox updates

```
DELETE { :A rdfs:subClassOf  :F . }
```

rewrite(u)

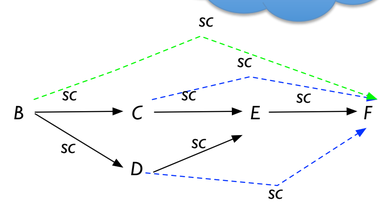
- Outbound cut

$Sem_{outcut}^{mat}$




```
DELETE { :A rdfs:subClassOf ?X. }
WHERE
{ :A rdfs:subClassOf ?X .
  ?X rdfs:subClassOf* :F. }
```

*Idea*: Can be implemented with SPARQL 1.1 property paths

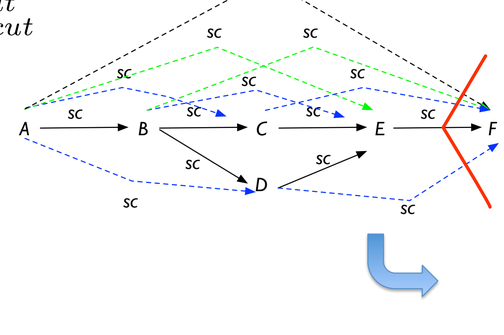



```
DELETE { :A rdfs:subClassOf  :F . }
```

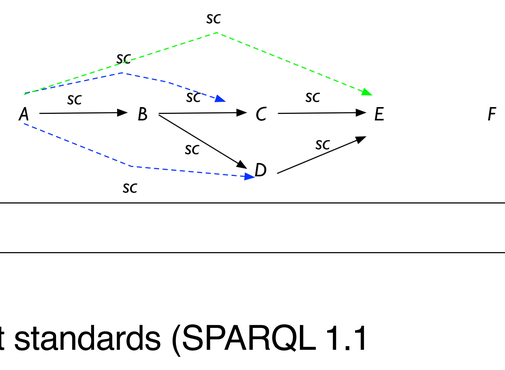
rewrite(u)

- Inbound cut

$Sem_{incut}^{mat}$




```
DELETE { ?X rdfs:subClassOf :F. }
WHERE
{ :A rdfs:subClassOf* ?X .
  ?X rdfs:subClassOf :F. }
```




## Red-preserving semantics: ABox updates

- $Sem_0^{red}$ … baseline semantics

$$G^{\mathbf{Sem}_0^{red}}_{u(P_d,P_i,P_w)} = reduce(G_{u(P_d,P_i,P_w)})$$

- $Sem_1^{red}$ … delete incl. causes

$$G^{\mathbf{Sem}_1^{red}}_{u(P_d,P_i,P_w)} = reduce(G_{u(P_d^{caus},P_i,\{rewrite(P_w)\}\{P_d^{fvars}\})})$$

$$P_d^{fvars} = \{\texttt{?x a rdfs:Resource.} \mid \text{for each } \texttt{?x} \in Var(P_d^{caus_{P_d}}) \setminus Var(P_d)\}$$

## Conclusion & Future work

- First step to close the gap left by the current standards (SPARQL 1.1 Update vs SPARQL1.1 Entailment Regimes)
- No “one-size fits all” semantics
- Non-intuitive corner cases in each semantics

**Future work:**

- Extend with OWL QL/RL features for expressing TBox
- Implementation and evaluation against different triple stores