WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

EFMD
EQUIS
ACCREDITED

Handling Inconsistencies due to Class Disjointness in SPARQL Updates

joint work with: Albin Ahmeti, Diego Calvanese, Vadim Savenkov

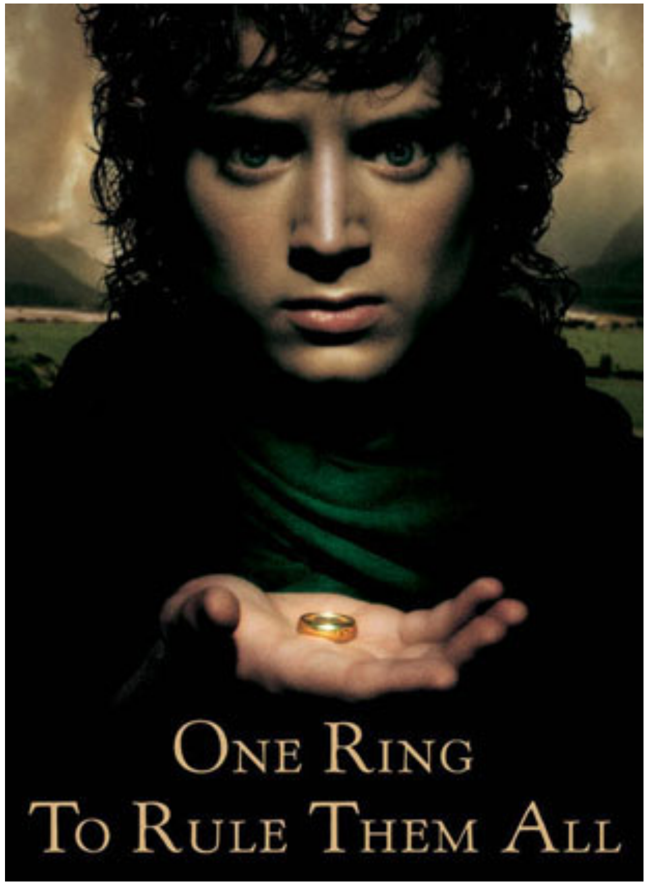
Axel Polleres

web: http://polleres.net **twitter: @AxelPolleres**

The quest...
One Semantics to rule them all

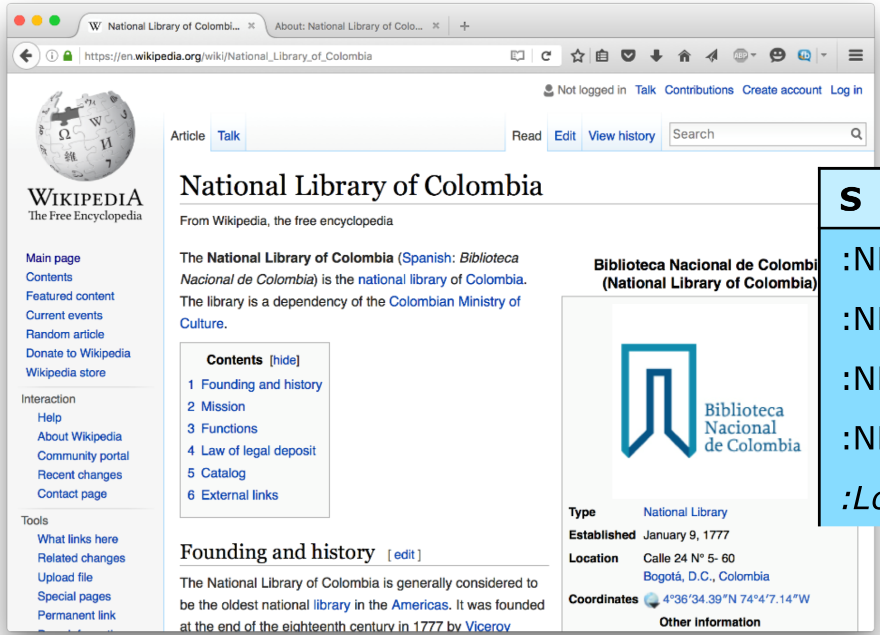
SPARQL1.1 Updates and Entailment - *Why the specification is silent about their interaction*

- **SPARQL1.1 Update** allows to update RDF Graphs
- **SPARQL1.1 Entailment Regimes** tells us what answers a SPARQL query gives us including implicit triples
- But: What does it mean to update implicit triples?
- *Particularly (in this paper): How to deal with inconsistencies?*



In which **library** do I find the "Lord of the Rings" and where do I find it?

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS



National Library of Colombi... | About: National Library of Colo...

https://en.wikipedia.org/wiki/National_Library_of_Colombia

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link

National Library of Colombia

From Wikipedia, the free encyclopedia

The **National Library of Colombia** (Spanish: *Biblioteca Nacional de Colombia*) is the national library of Colombia. The library is a dependency of the Colombian Ministry of Culture.

Contents [hide]
1 Founding and history
2 Mission
3 Functions
4 Law of legal deposit
5 Catalog
6 External links

Founding and history [edit]

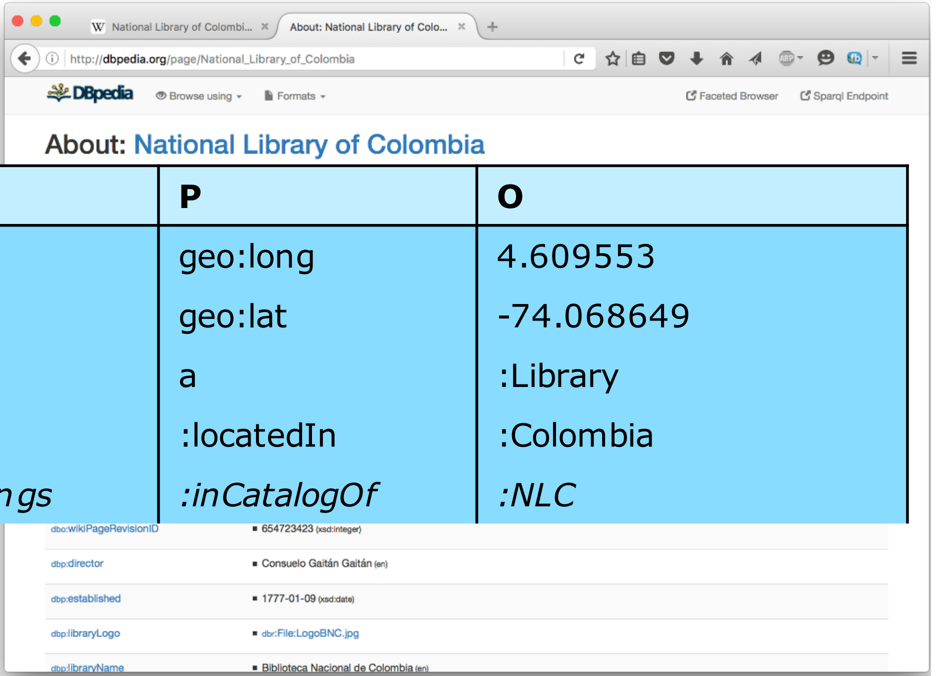
The National Library of Colombia is generally considered to be the oldest national library in the Americas. It was founded at the end of the eighteenth century in 1777 by Viceroy

Biblioteca Nacional de Colombi
(National Library of Colombia)

Biblioteca Nacional de Colombia

Type	National Library
Established	January 9, 1777
Location	Calle 24 N° 5- 60 Bogotá, D.C., Colombia
Coordinates	4°36′34.39″N 74°4′7.14″W

Other information



National Library of Colombi... | About: National Library of Colo...

http://dbpedia.org/page/National_Library_of_Colombia

DBpedia Browse using Formats Faceted Browser Sparql Endpoint

About: National Library of Colombia

S	**P**	**O**
:NLC	geo:long	4.609553
:NLC	geo:lat	-74.068649
:NLC	a	:Library
:NLC	:locatedIn	:Colombia
:LordOfTheRings	*:inCatalogOf*	*:NLC*

dbo:wikiPageRevisionID	▪ 654723423 (xsd:integer)
dbp:director	▪ Consuelo Gaitán Gaitán (en)
dbp:established	▪ 1777-01-09 (xsd:date)
dbp:libraryLogo	▪ dbr:File:LogoBNC.jpg
dbp:libraryName	▪ Biblioteca Nacional de Colombia (en)

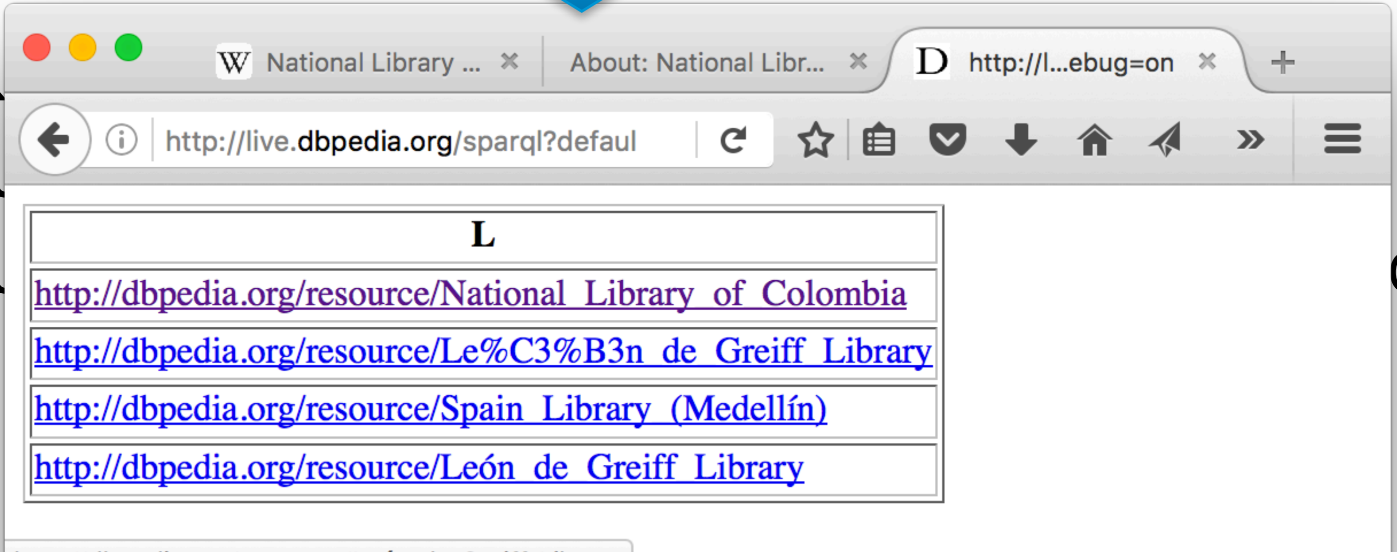
EFMD EQUIS ACCREDITED

SPARQL 1.1 Query language

- SPARQL offers a standard protocol/service interface to data offering services like DBPedia!

```
SELECT ?L WHERE {
:LordOfTheRings :inCatalogOf ?L.
?L a :Library .
}
```

- SPAR
  → Qu
  ← Qu ON, ...)



L
http://dbpedia.org/resource/National_Library_of_Colombia
http://dbpedia.org/resource/Le%C3%B3n_de_Greiff_Library
http://dbpedia.org/resource/Spain_Library_(Medellín)
http://dbpedia.org/resource/León_de_Greiff_Library

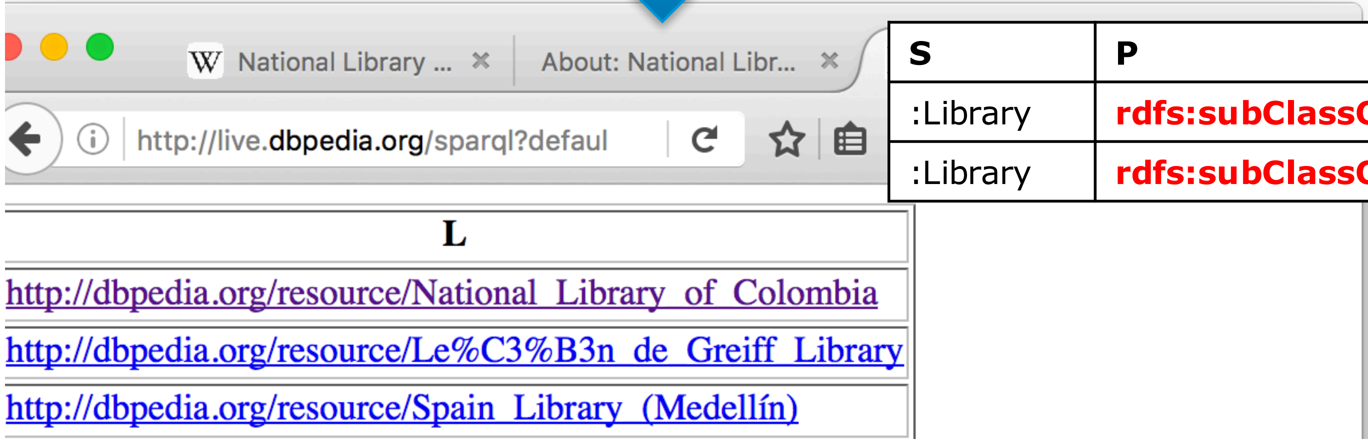
SPARQL1.1 Entailment Regimes:

- Make use of ontological infernces (**RDFS** and **OWL**):

```
SELECT ?L WHERE {
:LordOfTheRings :inCatalogOf ?L.
?L a schema:Library .
}
```

S	P	O
:NLC	geo:long	4.609553
:NLC	geo:lat	-74.068649
:NLC	a	:Library
:NLC	:locatedIn	:Colombis
:LordOfTheRings	:inCatalogOf	:NLC

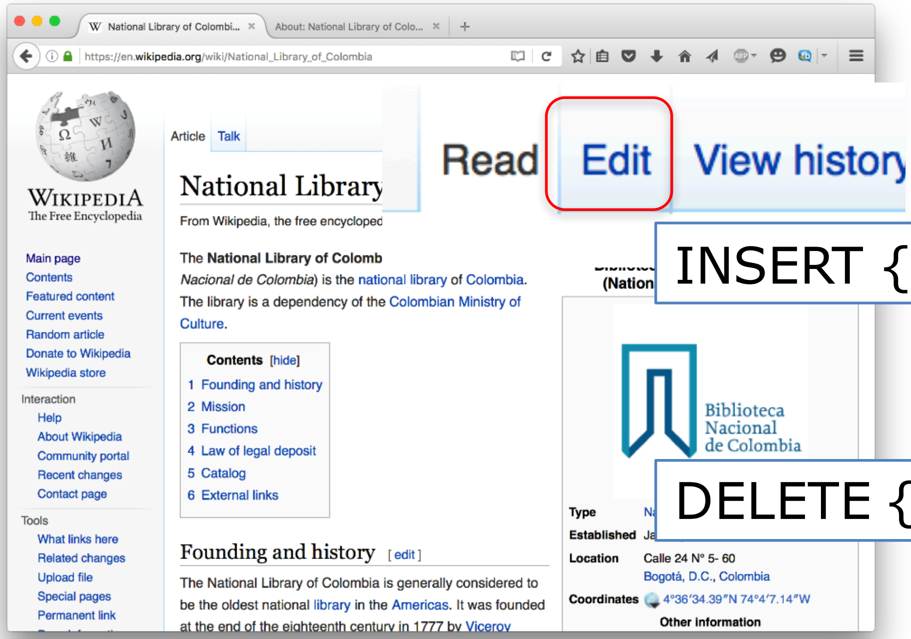
S	P	O
:Library	**rdfs:subClassOf**	:Organisation
:Library	**rdfs:subClassOf**	:schema:Library



L
http://dbpedia.org/resource/National_Library_of_Colombia
http://dbpedia.org/resource/Le%C3%B3n_de_Greiff_Library
http://dbpedia.org/resource/Spain_Library_(Medellín)

But: the Semantic Web is all about updates! SPARQL 1.1 Update

- What do updates mean?



Article Talk

Read Edit View history

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link

National Library

From Wikipedia, the free encycloped

The National Library of Colomb *Nacional de Colombia*) is the national library of Colombia. The library is a dependency of the Colombian Ministry of Culture.

Contents [hide]
1 Founding and history
2 Mission
3 Functions
4 Law of legal deposit
5 Catalog
6 External links

Founding and history [edit]

The National Library of Colombia is generally considered to be the oldest national library in the Americas. It was founded at the end of the eighteenth century in 1777 by Viceroy

(Nation

Biblioteca Nacional de Colombia

Type	Na
Established	Ja
Location	Calle 24 N° 5- 60 Bogotá, D.C., Colombia
Coordinates	4°36′34.39″N 74°4′7.14″W

Other information

```
INSERT { :NLC :a :Place.}
```

```
DELETE { :NLC a O[illegible].}
```

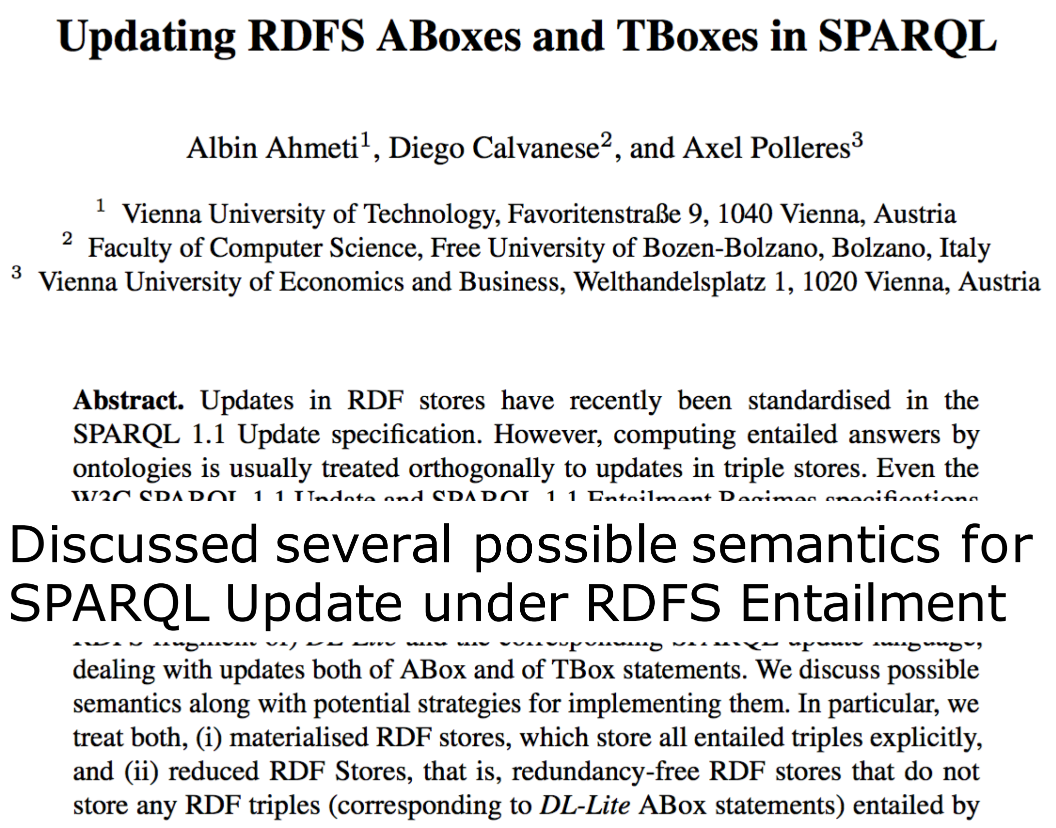
```
DELE[illegible]a :Organisation.}
INSER[illegible].X :a Place. }
WHERE { ?X geo:lat [] ; geo:long []. }
```

What do these updates mean in an RDF store that does Entailment?

Previous work: What happened before…

Our initial thoughts on this problem…

ISWC 2014
Riva del Garda - Trentino, Italy



Updating RDFS ABoxes and TBoxes in SPARQL

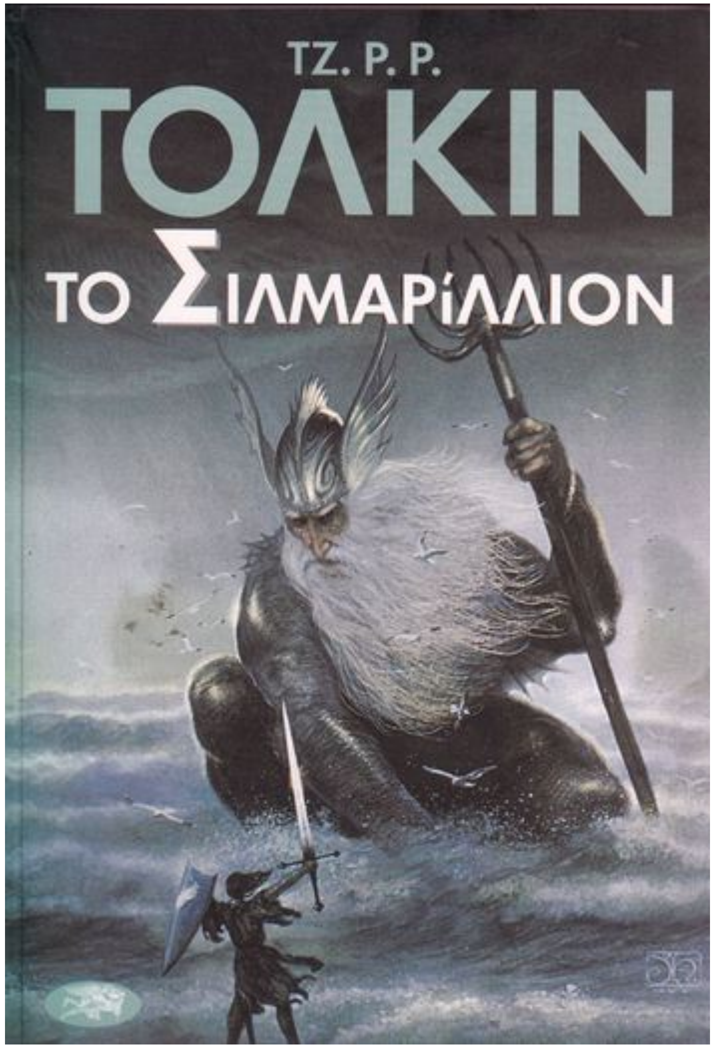
Albin Ahmeti[1], Diego Calvanese[2], and Axel Polleres[3]

[1] Vienna University of Technology, Favoritenstraße 9, 1040 Vienna, Austria
[2] Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy
[3] Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

Abstract. Updates in RDF stores have recently been standardised in the SPARQL 1.1 Update specification. However, computing entailed answers by ontologies is usually treated orthogonally to updates in triple stores. Even the W3C SPARQL 1.1 Update and SPARQL 1.1 Entailment Regimes specifications

RDFS fragment of) *DL-Lite* and the corresponding SPARQL update language, dealing with updates both of ABox and of TBox statements. We discuss possible semantics along with potential strategies for implementing them. In particular, we treat both, (i) materialised RDF stores, which store all entailed triples explicitly, and (ii) reduced RDF Stores, that is, redundancy-free RDF stores that do not store any RDF triples (corresponding to *DL-Lite* ABox statements) entailed by

Discussed several possible semantics for SPARQL Update under RDFS Entailment



Previous work: Our initial assumptions…

- Materialised store…

S	P	O
:NLC	geo:long	4.609553
:NLC	geo:lat	-74.068649
:NLC	a	:Library
:NLC	:hasDirector	:ConsueloGaitánGaitán
:LordOfTheRings	:inCatalogOf	:NLC
:NLC	**a**	**:Organisation**
:NLC	**a**	**schema:Library**

S	P	O
:Library	**rdfs:subClassOf**	:Organisation
:Library	**rdfs:subClassOf**	:schema:Library

- Low expressivity ontology language… RDFS

- Semantics for update should:
  - Preserve materialisation
  - Not "leave traces"

Previous work: Our initial solution...

- Idea: keep Materialised state by rewriting updates:

DELETE { P_d }
INSERT { P_i }
WHERE { P_w }

  - ***Sem$_2$^{mat}***
    - **Insert** the instantiations of P_i **plus all their effects.**
    - **Delete** the instantiations of P_d **plus all their causes**;

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

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

Our initial solution… Example:

```
DELETE {?X  :a :Organisation.}
INSERT { ?X :a Place. }
WHERE { ?X geo:lat [] ; geo:long []. }
```

rewrite(u,T)

```
DELETE {?X  :a :Organisation.  ?X a Library}
INSERT { ?X :a Place.  ?X a schema:Place}
WHERE { ?X geo:lat [] ; geo:long []. }
```

TBOX

S	P	O
:Library	**rdfs:subClassOf**	:Organisation
:Library	**rdfs:subClassOf**	schema:Library
:Place	**rdfs:SubClassOf**	schema:Place

ABOX

S	P	O
:NLC	geo:long	4.609553
:NLC	geo:lat	-74.068649
:NLC	a	:Library
:NLC	:hasDirector	:ConsueloGaitánGaitán
:LordOfTheRings	:inCatalogOf	:NLC
:NLC	a	:Organisation
:NLC	a	schema:Library

Our initial solution… Example:

```
DELETE {?X  :a :Organisation.}
INSERT { ?X :a Place. }
WHERE { ?X geo:lat [] ; geo:long []. }
```

rewrite(u,T)

```
DELETE {?X  :a :Organisation.  ?X a Library}
INSERT { ?X :a Place.  ?X a schema:Place}
WHERE { ?X geo:lat [] ; geo:long []. }
```

TBOX

S	**P**	**O**
:Library	**rdfs:subClassOf**	:Organisation
:Library	**rdfs:subClassOf**	schema:Library
:Place	**rdfs:SubClassOf**	schema:Place

ABOX

S	**P**	**O**
:NLC	geo:long	4.609553
:NLC	geo:lat	-74.068649
~~:NLC~~	~~a~~	~~:Library~~
:NLC	:hasDirector	:ConsueloGaitánGaitán
:LordOfTheRings	:inCatalogOf	:NLC
~~:NLC~~	~~a~~	~~:Organisation~~
:NLC	a	schema:Library
:NLC	**a**	**:Place**
:NLC	**a**	**schema:Place**

for many use cases the most 'reasonable", among the semantics we looked into…

Let’s revisit our initial assumptions…

- Materialised store…
  - … fits e.g. DBpedia (all Abox inferences are materialised)
  - **consistent**

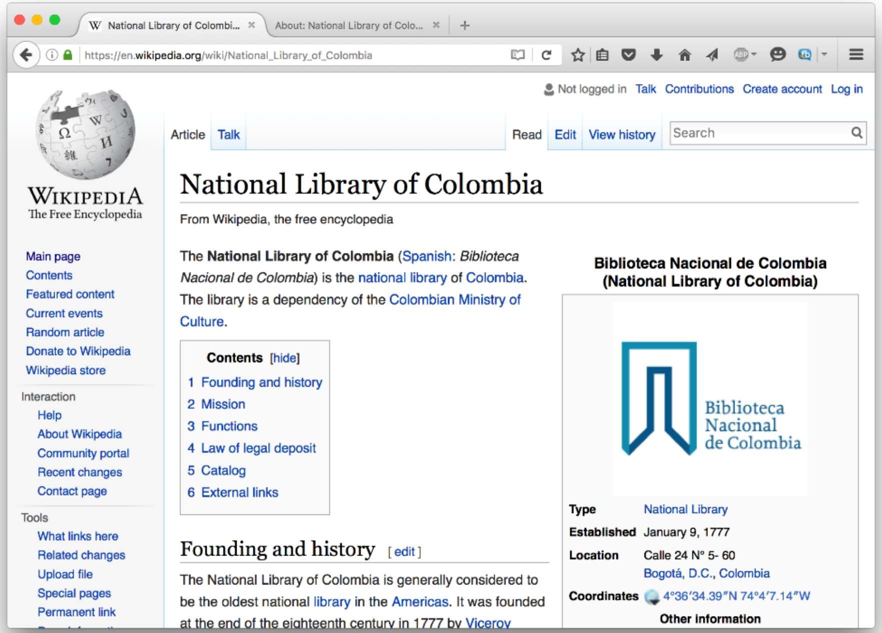
- Low expressivity ontology language… RDFS
  - ... does not quite fit DBpedia:
    - “OWL Dbpedia” :
    - rdfs:subClassOf, rdfs:subPropertyOf rdfs:domain, rdfs:range, **owl:inverseOf, owl:disjointWith**
- Semantics for update should:
  - Preserve materialisation
  - Not “leave traces”
  - **Preserve consistency**

Inconsistencies!

EFMD EQUIS ACCREDITED

Inconsistencies in DBPedia:

unfortunately **there are** inconsistencies in DBpedia…



National Library of Colombia

From Wikipedia, the free encyclopedia

The **National Library of Colombia** (Spanish: *Biblioteca Nacional de Colombia*) is the national library of Colombia. The library is a dependency of the Colombian Ministry of Culture.

Contents [hide]
1 Founding and history
2 Mission
3 Functions
4 Law of legal deposit
5 Catalog
6 External links

Founding and history [edit]

The National Library of Colombia is generally considered to be the oldest national library in the Americas. It was founded at the end of the eighteenth century in 1777 by Viceroy

Biblioteca Nacional de Colombia (National Library of Colombia)

Biblioteca Nacional de Colombia

Type	National Library
Established	January 9, 1777
Location	Calle 24 N° 5- 60 Bogotá, D.C., Colombia
Coordinates	4°36′34.39″N 74°4′7.14″W

Other information

S	P	O
:NLC	geo:long	4.609553
:NLC	geo:lat	-74.068649
:NLC	a	:Library
:NLC	:hasDirector	:ConsueloGaitánGaitán
:LordOfTheRings	:inCatalogOf	:NLC
:NLC	**a**	**:Organisation**
:NLC	a	schema:Library
:NLC	**a**	**:Place**

S	P	O
:Library	rdfs:subClassOf	:Organisation
:Library	rdfs:subClassOf	schema:Library
:Place	rdfs:SubClassOf	schemaPlace
:Place	**owl:disjointWith**	**:Organisation**

Can be introduced due to uncautious **updates** and the flexibility of mappings ☹

So, how can we do SPARQL updates that preserve consistency (and materialization?

- Dealing with different forms of inconsistencies:
  - Intrinsic inconsistencies "within" updates
  - Inconsistencies between "old" and "new" knowledge
    - … Different solution strategies:
      - Brave
      - Cautious
      - Fainthearted (somewhere in between ;-))

So, how can we do SPARQL updates that preserve consistency (and materialization?

- Dealing with different forms of inconsistencies:
  - **Intrinsic inconsistencies "within" updates**
    - **… solution: "safe" rewriting**
  - Inconsistencies between "old" and "new" knowledge
    - … Different solution strategies:
      - Brave
      - Cautious
      - Fainthearted (somewhere in between ;-))

SPARQL updates: deal with inconsistency within ***new knowledge***

:Place owl:disjointWith :Organisation . $Place \sqsubseteq \neg Organisation$

:based_near rdfs:domain :Organisation. $\exists based\_near \sqsubseteq Organisation$

:based_near rdfs:range :Place. $\exists based\_near^{-} \sqsubseteq Place$

***"Unsafe"* update → intrinsically inconsistent**:

INSERT {?X :based_near ?Y } WHERE { ?X :locatedIn ?Y .}

:NLC :locatedIn :Bogotá .

:Bogotá :locatedIn Colombia .

SPARQL updates: deal with inconsistency within ***new knowledge***

:Place owl:disjointWith :Organisation .

:based_near rdfs:domain :Organisation.

:based_near rdfs:range :Place.

***"Unsafe"* update**:

```
INSERT {?X :based_near?Y}  WHERE {?X :locatedIn ?Y .}
```

intrinsical Inconsistencies can be caught by "**safe rewriting**"

```
INSERT{?X :based_near ?Y}
WHERE{?Y  :locatedIn ?X .
       MINUS{  {?X1 :locatedIn ?Y}
               UNION {?X :locatedIn ?Y2}}}
```

Copies of the WHERE clause, variables renamed appropriately.

Safe Rewriting – for the SPARQL enthusiasts: MINUS vs. FILTER NOT EXISTS

Safe rewriting via FILTER NOT EXISTS doesn't work (Corner case example):

```
DELETE{?Z a :Place} INSERT{?X :locatedIn ?Y}
WHERE{  {?Y :locatedIn ?X}
             UNION {?V :locatedIn ?Z}}
```

Safe Rewriting – for the SPARQL enthusiasts: MINUS vs. FILTER NOT EXISTS

Safe rewriting via FILTER NOT EXISTS doesn't work:

```
DELETE{?Z a :Place} INSERT{?X :locatedIn ?Y}
WHERE{  {?Y :locatedIn ?X}
          UNION {?V :locatedIn ?Z}}
FILTER NOT EXISTS{
{?X1  :locatedIn?Y}  UNION {?V1 :locatedIn?Z1}
}}
FILTER NOT EXISTS{
{?X :locatedIn?Y2}  UNION {?V2 :locatedIn ?Z2}
}}
```

disjoint sets of variables

Exists whenever the WHERE clause is satisfied!

Simply renaming the whole WHERE clause is not possible.

Safe Rewriting – for the SPARQL enthusiasts: MINUS vs. FILTER NOT EXISTS

Safe rewriting via MINUS: works!

```
DELETE{?Z a :Place} INSERT{?X :locatedIn ?Y}
WHERE{  {?Y :locatedIn ?X}
            UNION {?V :locatedIn ?Z}}
MINUS{
{?X1  :locatedIn?Y}  UNION {?V1 :locatedIn?Z1}
}}
MINUS{
{?X :locatedIn?Y2}  UNION {?V2 :locatedIn?Z2}
}}
```

Extra union branches do not matter!

- MINUS removes variable bindings of the WHERE clause *that can be combined with **some** result of the query in its right-hand side*.
- Only variables from the left-hand side of MINUS are "visible" in ist right-hand side: great for our case!

So, how can we do SPARQL updates that preserve consistency (and materialization?

- Dealing with Different forms of inconsistencies:
  - Intrinsic inconsistencies within updates
    - … solution: “safe” rewriting
  - **Inconsistencies between “old” and “new” knowledge**
    - **… Different solution strategies:**
      - **Brave**
      - **Cautious**
      - **Fainthearted (somewhere in between ;-))**

SPARQL updates: deal with inconsistency w.r.t. the old knowledge

Idea: *adapt* $\boldsymbol{Sem_2^{mat}}$ (rewriting-based) semantics

- **Brave**: when in conflict, prefer new knowledge
  - cf. FastEvol [Calvanese et al 2010]

- **Cautious**: when in conflict, stick to the old knowledge
  - In batch updates, allow variable bindings only where the insert clause does not produce a clash

- **Fainthearted**: relaxation of cautious semantics
  - the same batch update might resolve clashes by deleting conflicting parts of the old knowledge!

Example: Brave ***Sem***$_2^{mat}$

:Place owl:disjointWith :Organisation .

:based_near rdfs:domain :Organisation.

:based_near rdfs:range :Place.

```
INSERT{?X :based_near?Y} WHERE{?Y :locatedIn ?X}
```

Preprocess:

safe rewriting

```
INSERT{?X :based_near?Y}
WHERE{?Y :locatedIn ?X
                MINUS{  {?Y1 :locatedIn?X}
                        UNION {?Y :locatedIn?X2}}}
```

Example: Brave $\textbf{\textit{Sem}}_2^{\textit{mat}}$

:Place owl:disjointWith :Organisation .

:based_near rdfs:domain :Organisation.

:based_near rdfs:range :Place.

INSERT{?X : based_near ?Y} WHERE{?X :locatedIn ?Y}

Brave $\textbf{\textit{Sem}}_2^{\textit{mat}}$

```
DELETE {?X a :Place . ?X3 :based_near ?X .
           ?Y a :Organization. ?Y :based_near ?Y3 }
INSERT{?X :based_near ?Y . ?X  a :Organisation . ?Y a :Place}
WHERE{?X :locatedIn ?Y
              MINUS{  {?Y1 :locatedIn ?X}
                        UNION {?Y :locatedIn ?Y2}}
              OPTIONAL {?X3 :based_near ?X}
              OPTIONAL {?Y :based_near?Y3}}
```

Potential clashes

Bind variables in DELETE

Example: Cautious $\textbf{\textit{Sem}}_2^{\textbf{\textit{mat}}}$

:Place owl:disjointWith :Organisation .

:based_near rdfs:domain :Organisation.

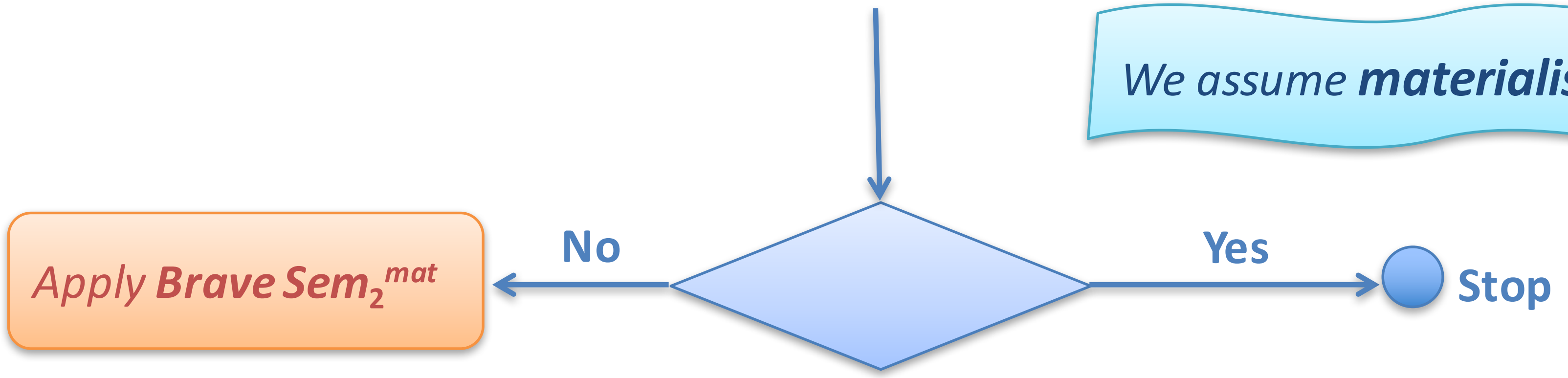
:based_near rdfs:range :Place.

INSERT{?X :based_near ?Y} WHERE{?X :locatedIn ?Y}

Cautious $\textbf{\textit{Sem}}_2^{\textbf{\textit{mat}}}$

ASK WHERE{?X :locatedIn ?Y .
{{?X :a :Place} UNION {?Y :a :Organization}} }

*We assume **materialised** store!*



Example: Cautious ***Sem***$_2^{\textbf{\textit{mat}}}$

Removes some clashes!

wl:disjointWith :Organisation .

near rdfs:domain :Organisation.

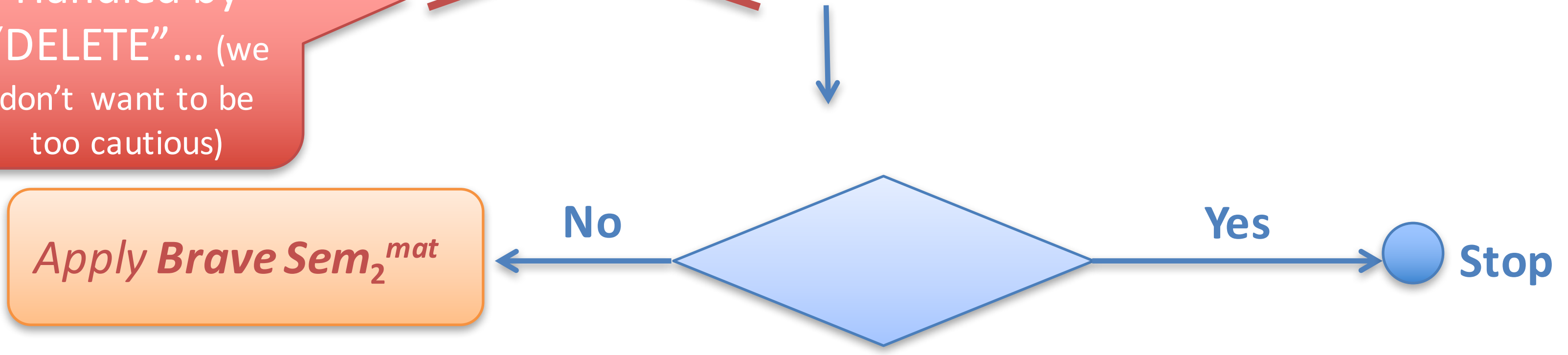
based_near rdfs:range :Place.

DELETE{?X :Place}
INSERT{?X :based_near ?Y} WHERE{?X :locatedIn ?Y}

Cautious Sem$_2^{\textbf{\textit{mat}}}$

ASK WHERE{?X :locatedIn ?Y .
{~~{?X :a :Place}~~ UNION {?Y :a :Organization}} }

Handled by "DELETE"… (we don't want to be too cautious)



Finally:

Example: Fainthearted ***Sem***$_2^{mat}$

:Place owl:disjointWith :Organisation .

Do inserts only with non-clashing variable bindings

DELETE{?X :Place}
INSERT{?X :based_near ?Y} WHERE{?X :locatedIn ?Y}

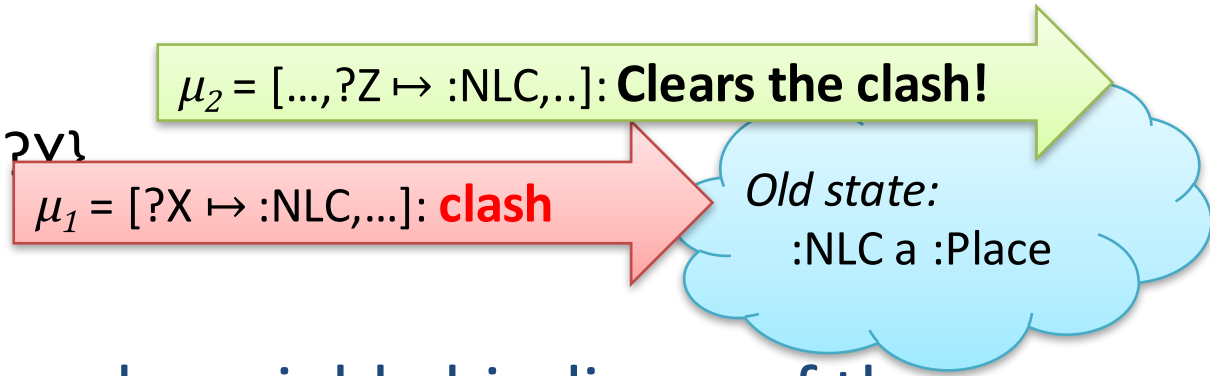
Fainthearted Sem$_2^{mat}$

```
DELETE{?X :Place . }
INSERT{?X :based_near ?Y . ?X  a :Organisation . ?Y a :Place}
WHERE{?X :locatedIn ?Y MINUS{  {?Y1 :locatedIn ?X}
                                UNION {?Y :locatedIn ?Y2}}
        MINUS {~~{?X a :Place} UNION~~ {?Y a :Organisation}}}
```

Again, handled by "DELETE"

Fainthearted semantics: pitfalls, e.g. clashes removed by **different** bindings

```
DELETE {?Z a :Place}
INSERT {?X :based_near ?Y}
WHERE { … }
```

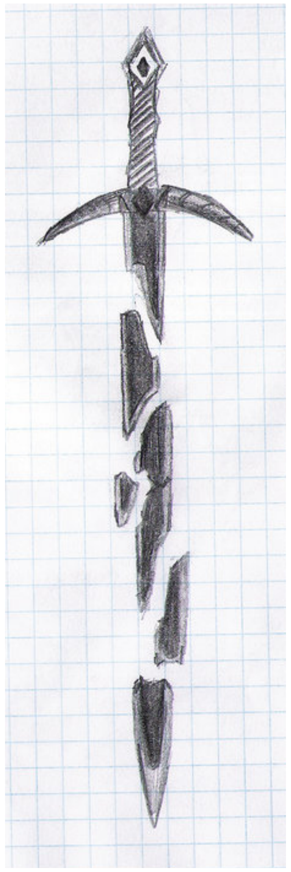

μ_2 = […,?Z ↦ :NLC,..]: **Clears the clash!**
μ_1 = [?X ↦ :NLC,…]: **clash**
Old state:
:NLC a :Place

- Atomic updates: for each variable binding μ of the WHERE clause either **both** delete **and** insert or **none**.
- Insert with μ_1 depends on the deletion with μ_2... our initial approach would be too cautious.
- By atomiticy, if μ_2 also causes insertion (which might depend on the deletion by some μ_3, etc).

Idea: give up on update atomicity. Delete for all μ_i of the WHERE pattern, insert only where not clashing; for this we have to "separate" DELETE and INSERT... More involved rewriting → paper

Putting the pieces together: What else you find in the paper

- Details, general rewriting algorithms for Brave, Cautious and Feinthearted Update Semantics
- Experiment on some updates with LUBM50 (to show feasibility)
  - no clear winner in terms of performance...
  - optimizations are on our agenda.



- Working **prototype**, in principle pluggable on top of arbitrary SPARQL engines, available at:

http://dbai.tuwien.ac.at/user/ahmeti/sparqlupdate-inconsistency-resolver/

What's next?

- SPARQL Update + Entailments
  - from the ***"one ring"*** to the ***"Holy Grail":***
  - SPARQL Updates for full OBDA? (i.e. incl mappings)

- Initial work to extend our work to updates over DBPedia **including mappings** - forthcoming!

  (sneak preview: short paper at AMW2016, next week, Panama)