Semantics of RDF – S

Gilles Falquet
Semantic Web Technologies
2013

RDF semantics - Goals

- Evaluate the truth of a triple / graph
- Characterize the state of the world that make a triple / graph true

Defining the interpretation of a vocabulary

- Map each entity of the vocabulary (URI reference, literal) to a real-world object
- If an entity is a property then is is also mapped to a binary relation between real-world objects

More formally

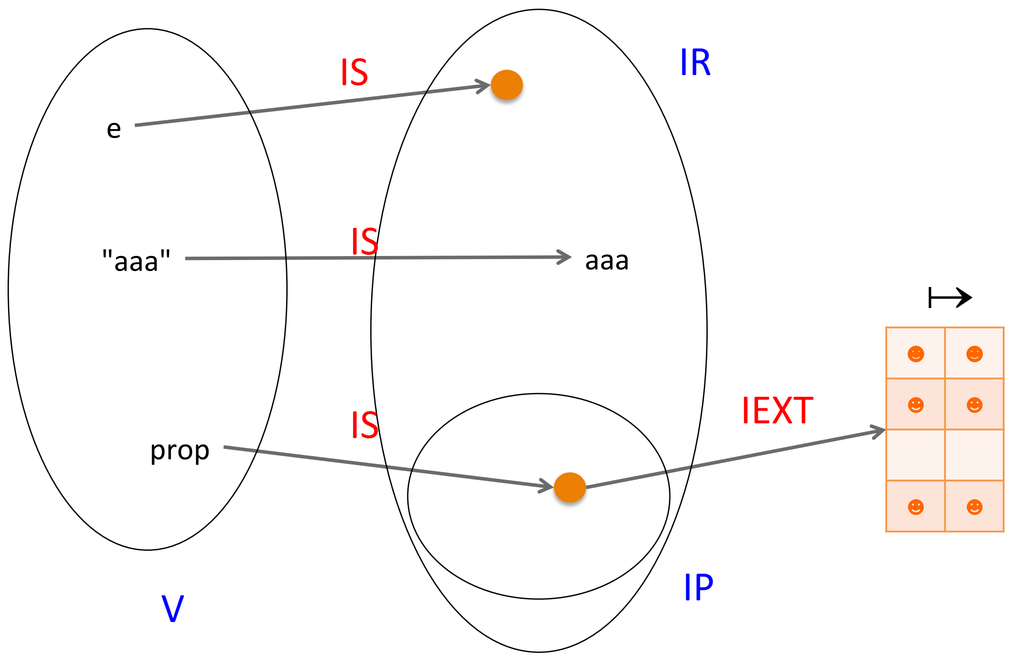
Interpretation relative to a vocabulary V

IR : interpretation domain = (set of all resources)

IP $\subseteq$ IR, interpretation domain for properties

Interpretation functions

- IS maps each URI reference of V to a resource of IR $\cup$ IP
- IL maps each literal value of V to a resource of IR
- IEXT maps each element of IP (property interpretation) to a relation in IR $\times$ IR (a set of resource pairs)



Exemple

V = {`http://cui.unige.ch/z.html`, `mailto:toto@unige.ch`, `mailto:bob@unil.ch`, `dc:author`}

$\mathrm{IR} = \mathrm{CUIDocs} \cup \mathrm{CUIMembers} \cup \{\mathrm{author}\}$

$\mathrm{IP} = \{\mathrm{author}\}$

IS(`http://cui.unige.ch/z.html`) = doc3

IS(`mailto:toto@unige.ch`) = member3

IS(`mailto:bob@unil.ch`) = member1

IS(`dc:author`) = author

IEXT(author) = {(doc1, member7), (doc3, member1), (doc3, member3)}}

Interpretation of a triple

I([s, p, o .]) = true
if an only if
$IEXT(IS(\text{p}))$ contains $(IS(\text{s}), IS(\text{o}))$

notation
$I \vDash$ [s, p, o .]

A graph is true (for a given interpretation) iff each triple is true

Example

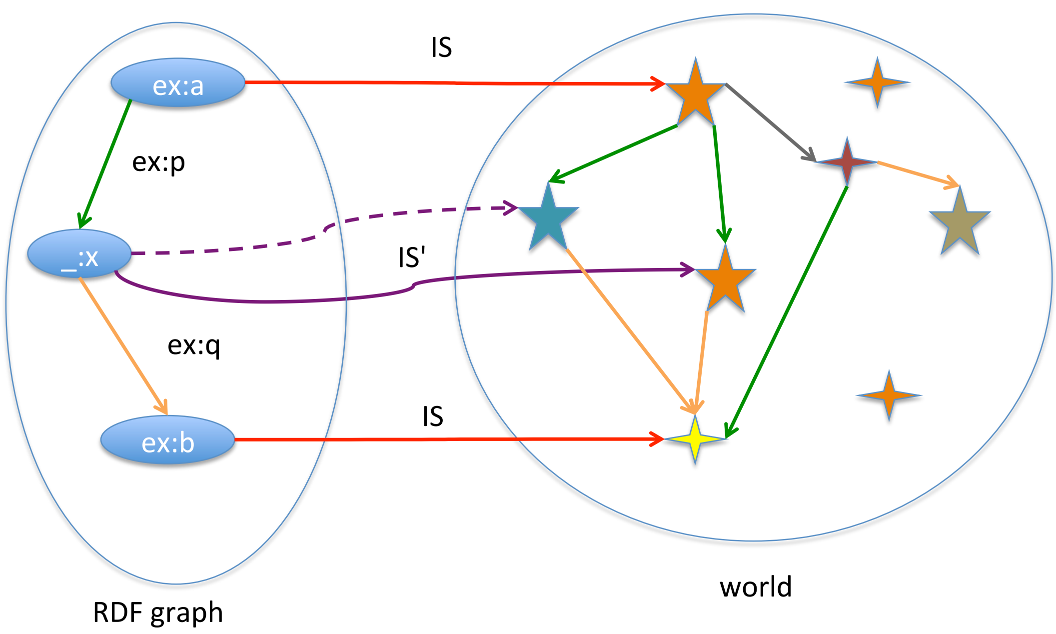
With the interpretation
IS(`http://cui.unige.ch/z.html`) = doc3
IS(`mailto:toto@unige.ch`) = member3
IS(`mailto:bob@unil.ch`) = member1
IS(`dc:author`) = author

$IEXT$(author) = {(doc1, member7), (doc1, member1), (doc3, member3)}}

`<http://cui.unige.ch/z.html> dc:author <mailto:toto@unige.ch>` is true
`<http://cui.unige.ch/z.html> dc:author <mailto:bob@unige.ch>` is false
`<mailto:toto@unige.ch> dc:author dc:author` is false

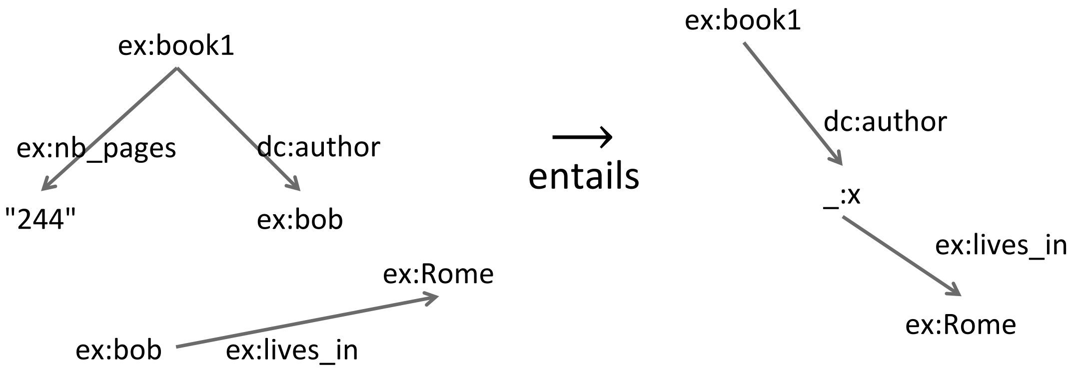
The meaning of blank nodes

- A blank node represents the existence of some resource
- A graph with blank nodes is true is
  - there is an extension IS' of IS to blank nodes
  - the graph is true for IS'



Graph entailment

- The usual definition of entailment
- $\{E_1, E_2, \ldots, E_k\}$ entails F if every interpretation which satisfies E_1 and E_2 and ...and E_k also satisfies E



Vocabulary interpretation and entailment

In simple interpretations and simple entailment no attention is paid to the particular meaning of any of the names in the graph.

To obtain the full meaning of an RDF graph written using a particular vocabulary

- add further semantic conditions which attach stronger meanings to particular URI references and typed literals in the graph.
- *vocabulary interpretations* = interpretations which are required to satisfy extra semantic conditions
- Vocabulary entailment means entailment with respect to such vocabulary interpretations. *rdf-entailment*, *rdfs-entailment*.

rdf-interpretation

An rdf-interpretation must satisfy semantic conditions

- x is in IP if and only if <x, I(rdf:Property)> is in IEXT(I(rdf:type))
  (something interpreted as a property must be of type Property)
- conditions on the literal values

It must also entail a set of "axiomatic triples":

```
rdf:type rdf:type rdf:Property .
rdf:subject rdf:type rdf:Property .
rdf:predicate rdf:type rdf:Property .
rdf:object rdf:type rdf:Property .
rdf:first rdf:type rdf:Property .
rdf:rest rdf:type rdf:Property .
rdf:value rdf:type rdf:Property .
rdf:_1 rdf:type rdf:Property .
rdf:_2 rdf:type rdf:Property .
...
rdf:nil rdf:type rdf:List .
```


remark

```
_:xxx rdf:type rdf:Bag .
_:xxx rdf:_1 <ex:a> .
_:xxx rdf:_2 <ex:b> .
```

does not entail

```
_:xxx rdf:_1 <ex:b> .
_:xxx rdf:_2 <ex:a> .
```

rdfs-interpretations

- Must take into account the rdfs vocabulary
  rdfs:domain rdfs:range rdfs:Resource rdfs:Literal rdfs:Datatype rdfs:Class rdfs:subClassOf rdfs:subPropertyOf rdfs:member rdfs:Container rdfs:ContainerMembershipProperty rdfs:comment rdfs:seeAlso rdfs:isDefinedBy rdfs:label

- In particular they must satisfy semantic conditions such as
  - `subClassOf` is transitive and reflexive for classes
  - `subPropertyOf` is transitive and reflexive for properties
  - If <x,y> is in IEXT(I(`rdfs:domain`)) and <u,v> is in IEXT(x) then u is a member of class y
  - etc.

Interpretation of a class

A class is usually interpreted as a set (the set of its instances)

Impossible with RDF since

```
x:A rdf:type x:B .
x:B rdf:type x:A .
```

is a legal RDF graph

With a set-based interpretation we would have

$$I(A) \in I(B) \in I(A) \in I(B) \in \ldots$$

But infinite chains of $\in$ are forbidden in set theory (foundation axiom)

Interpretation of a class

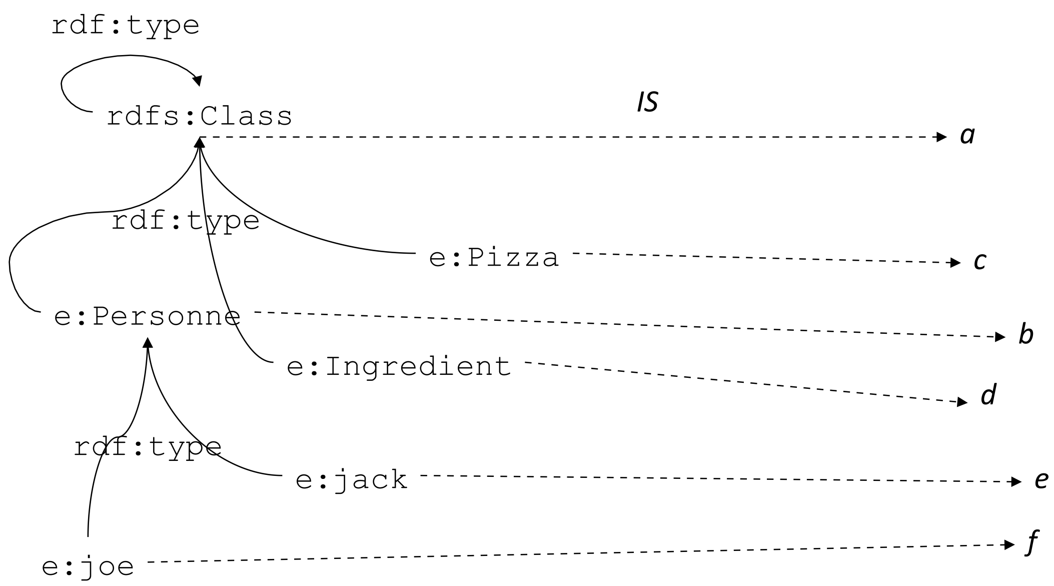
Introduce a class extension function : $ICEXT$(class)

IS(x) belongs to ICEXT(c) iff

(IS(x), IS(c)) is in $IEXT$(IS(`rdf:type`))

No problem with set theory, the interpretation of a class is a set of resources that are not sets.

[`rdfs:Class rdf:type rdfs:Class .`] entails
IS(`rdfs:Class`) ∈ $ICEXT$(`rdfs:Class`) but not
IS(`rdfs:Class`) ∈ IS(`rdfs:Class`)



ICEXT(`rdfs:Class`) = {*a, b, c, d*}

ICEXT(`e:Personne`) = {*e, f*}

Some RDFS axiomatic triples

```
rdf:type rdfs:range rdfs:Class .

rdfs:domain rdfs:range rdfs:Class .
rdfs:range rdfs:range rdfs:Class .

rdfs:subPropertyOf rdfs:range rdf:Property .
rdfs:subClassOf rdfs:range rdfs:Class .

rdf:subject rdfs:range rdfs:Resource .
rdf:predicate rdfs:range rdfs:Resource .
rdf:object rdfs:range rdfs:Resource .

rdfs:member rdfs:range rdfs:Resource .
rdf:first rdfs:range rdfs:Resource .
rdf:rest rdfs:range rdf:List .
```


Some RDFS axiomatic triples

```
rdf:Alt rdfs:subClassOf rdfs:Container .
rdf:Bag rdfs:subClassOf rdfs:Container .
rdf:Seq rdfs:subClassOf rdfs:Container .
rdfs:ContainerMembershipProperty rdfs:subClassOf rdf:Property .


rdfs:isDefinedBy rdfs:subPropertyOf rdfs:seeAlso .

rdf:XMLLiteral rdf:type rdfs:Datatype .
rdf:XMLLiteral rdfs:subClassOf rdfs:Literal .

rdfs:Datatype rdfs:subClassOf rdfs:Class .
```