RDFS Entailment

G. Falquet

Semantic web technologies

2014

- RDF is intended for use as a base notation for a variety of extended notations such as RDFS, OWL, RIF, ... whose expressions can be encoded as RDF graphs which use a particular vocabulary with a specially defined meaning. [1]

```
# RDFS
:Pizza a rdfs:Class
:VegPizza rdfs:subClassOf :Pizza
```

```
# OWL
:VegPizza rdf:type owl:Class ;
    owl:equivalentClass [ rdf:type
                owl:Restriction ;
                owl:onProperty :hasTopping ;
                owl:allValuesFrom :VegTopping
                ]
```

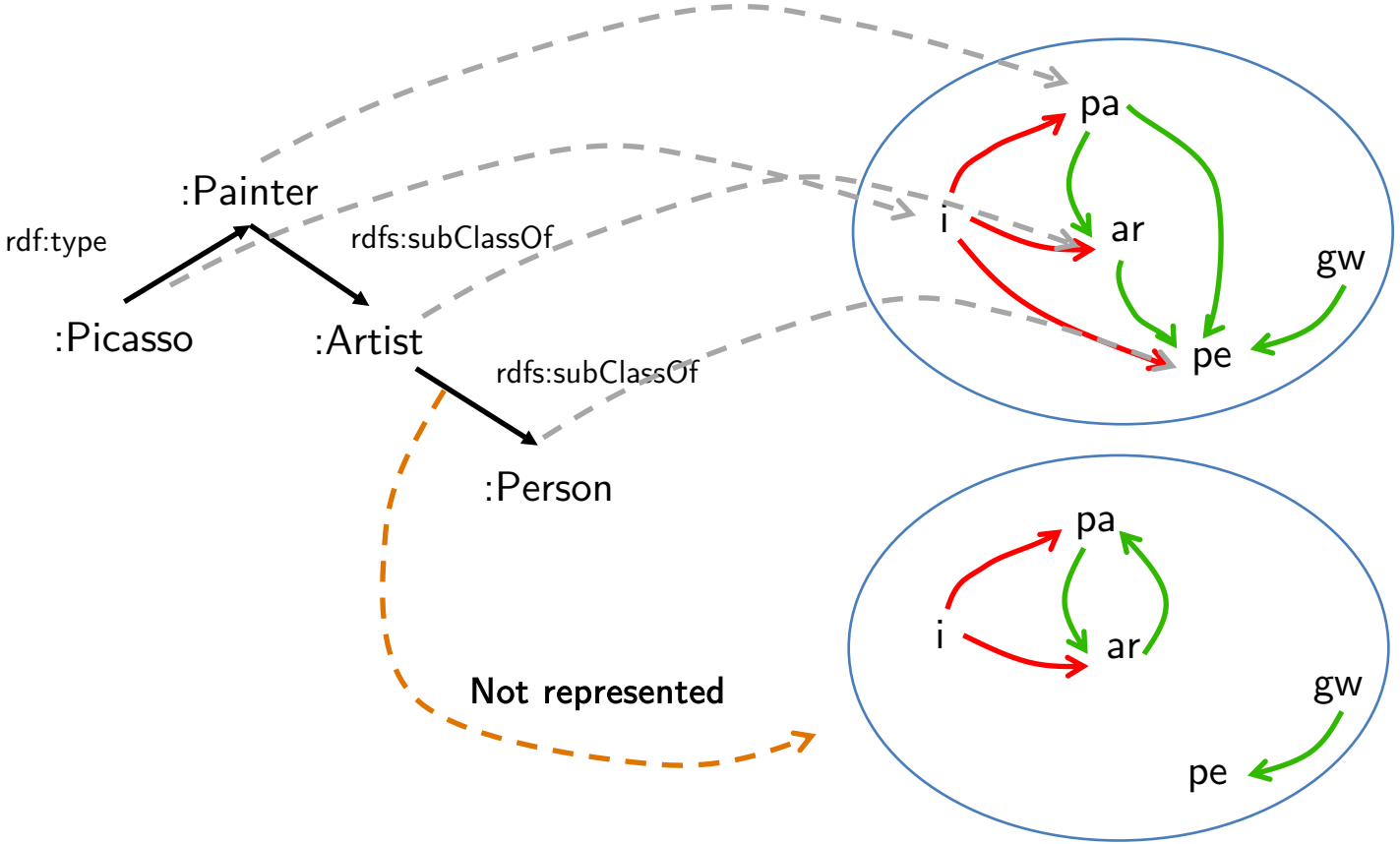
1. https://www.w3.org/TR/rdf11-mt/#entailment-rules-informative

Semantics

- For each notation there is a notion of **interpretation**
  - associates IRIs and blank nodes to domain objects
  - associates literals to values in a datatype domain
  - associates the interpretation of properties to binary relations over domain objects (**extensions**)

- An interpretation a graph is *true* if it satisfies
  - some semantic conditions
    - e.g. the extension of the interpretation of rdfs:subClassOf is a transitive relation
  - some axiomatic triples

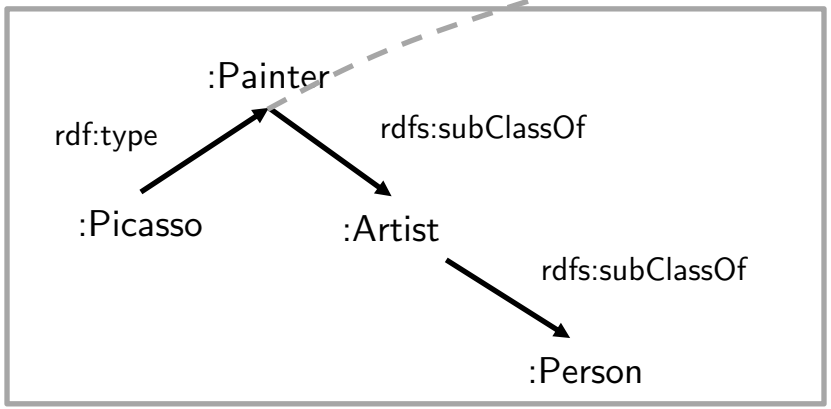
RDF Interpretations

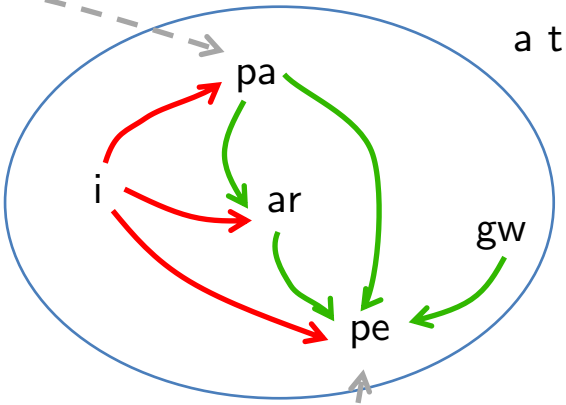


a true interpretation

a false interpretation

E

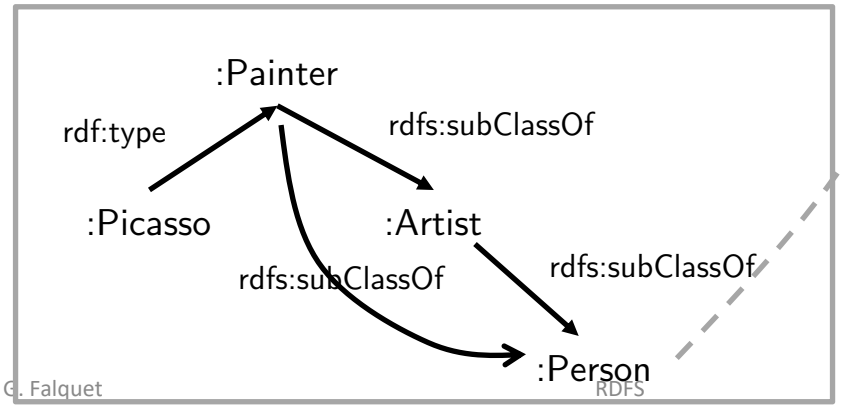




a true interpretation of E

also a true interpretation of F

F



Entailment

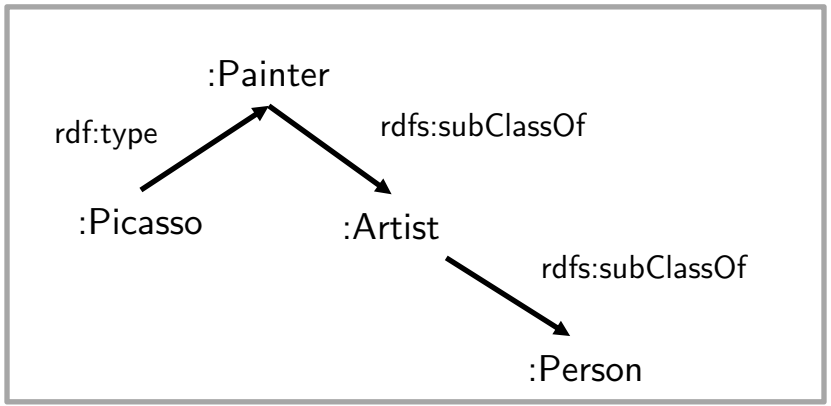
A graph E *N*-entails a graph F iff

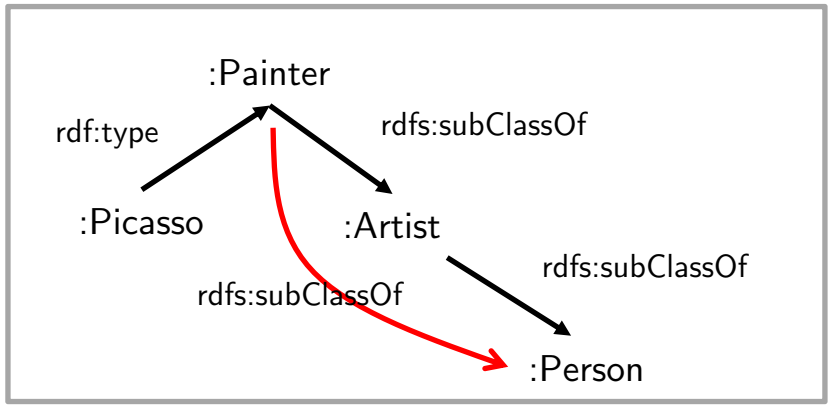
Each true *N*-interpretation of E is also a true *N*-interpretation of F.

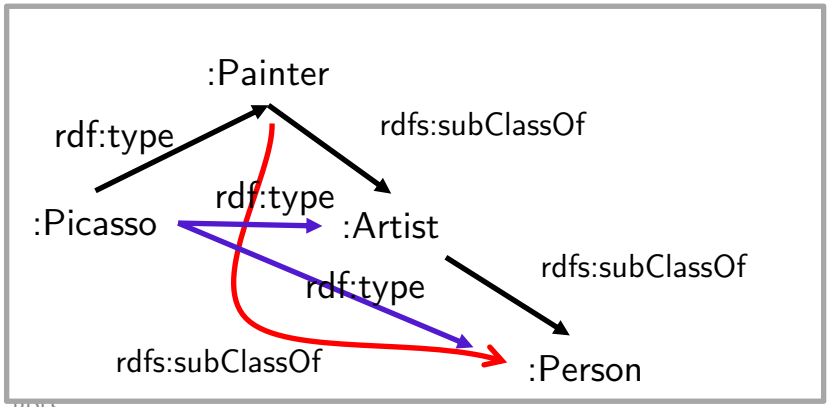
(*N* is a notation such as RDF, RDFS, OWL, ...)

= The usual notion of logical consequence

RDFS-Entailments







Computing RDFS-Entailment

RDFS entailment can be computed by

1. adding the axiomatic triples to the graph
2. applying inference patterns

Some axiomatic triples

```
rdf:type rdfs:domain rdfs:Resource .
rdfs:domain rdfs:domain rdf:Property .
rdfs:range rdfs:domain rdf:Property .
rdfs:subPropertyOf rdfs:domain rdf:Property .
rdfs:subClassOf rdfs:domain rdfs:Class .

rdf:first rdfs:domain rdf:List .
rdf:rest rdfs:domain rdf:List .
rdfs:seeAlso rdfs:domain rdfs:Resource .
rdfs:isDefinedBy rdfs:domain rdfs:Resource .
rdfs:comment rdfs:domain rdfs:Resource .
rdfs:label rdfs:domain rdfs:Resource .
rdf:value rdfs:domain rdfs:Resource .

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

Inference patterns (rules)

	If S contains:	**then S RDFS entails recognizing D:**
rdfs1	any IRI t in D	t rdf:type rdfs:Datatype .
rdfs2	p rdfs:domain x . y p z .	y rdf:type x .
rdfs3	p rdfs:range x . y p z .	z rdf:type x .
rdfs4a	x p y .	x rdf:type rdfs:Resource .
rdfs4b	x p y.	y rdf:type rdfs:Resource .
rdfs5	x rdfs:subPropertyOf y . y rdfs:subPropertyOf z .	x rdfs:subPropertyOf z . (transitivity)
rdfs6	x rdf:type rdf:Property .	x rdfs:subPropertyOf x . (reflexivity)

(cont)

	If S contains:	then S RDFS entails recognizing D:
rdfs6	x rdf:type rdf:Property .	x rdfs:subPropertyOf x . (reflexivity)
rdfs7	p rdfs:subPropertyOf q . x p y .	x q y .
rdfs8	x rdf:type rdfs:Class .	x rdfs:subClassOf rdfs:Resource .
rdfs9	x rdfs:subClassOf y . z rdf:type x .	z rdf:type y .
rdfs10	x rdf:type rdfs:Class .	x rdfs:subClassOf x . (reflexivity)
rdfs11	x rdfs:subClassOf y . y rdfs:subClassOf z .	x rdfs:subClassOf z . (transitivity)
rdfs12	x rdf:type rdfs:ContainerMembershipProperty .	x rdfs:subPropertyOf rdfs:member .
rdfs13	x rdf:type rdfs:Datatype .	x rdfs:subClassOf rdfs:Literal .

Example

:q rdfs:range :d .
:p rdfs:subPropertyOf :q .
:d rdfs:subClassOf e .
:a :p :b

RDFS Entails

:a :q :b
:b rdf:type :d
:b rdf:type :e

The rules are not complete

```
:p rdfs:subPropertyOf _:b .
_:b rdfs:domain :c .
:d :p :e .
```

entails

```
:d rdf:type :c .
```

But cannot be obtained by applying the rules

rdfs7 would produces

```
:d _:b :e
```

which is not legal in RDF (blanks not allowed as predicates)

The rules become complete on generalized RDF graphs with

- blanks allowed as predicates
- literals allowed as subjects

Entailment and tools

- Triple stores
  - may automatically generate the entailed triples when new triples are added
  - and retract them when triples are removed
  - the entailment regime is usually selected at repository creation

- Reasoners
  - tools that perform entailment (or other reasoning tasks) on existing graphs

- SPARQL engines
  - either make use of the entailed triples
  - or have call a reasoner before (or while) executing queries

Entailment and Other Vocabularies

- The shared vocabularies may contain rdf triples that can be used in entailments
- A vocabulary must be physically imported into the working graph (there is no "import" statement in RDF)