Cours 2 + 3

RDFS Entailment

G. Falquet

Semantic web technologies

- 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
```

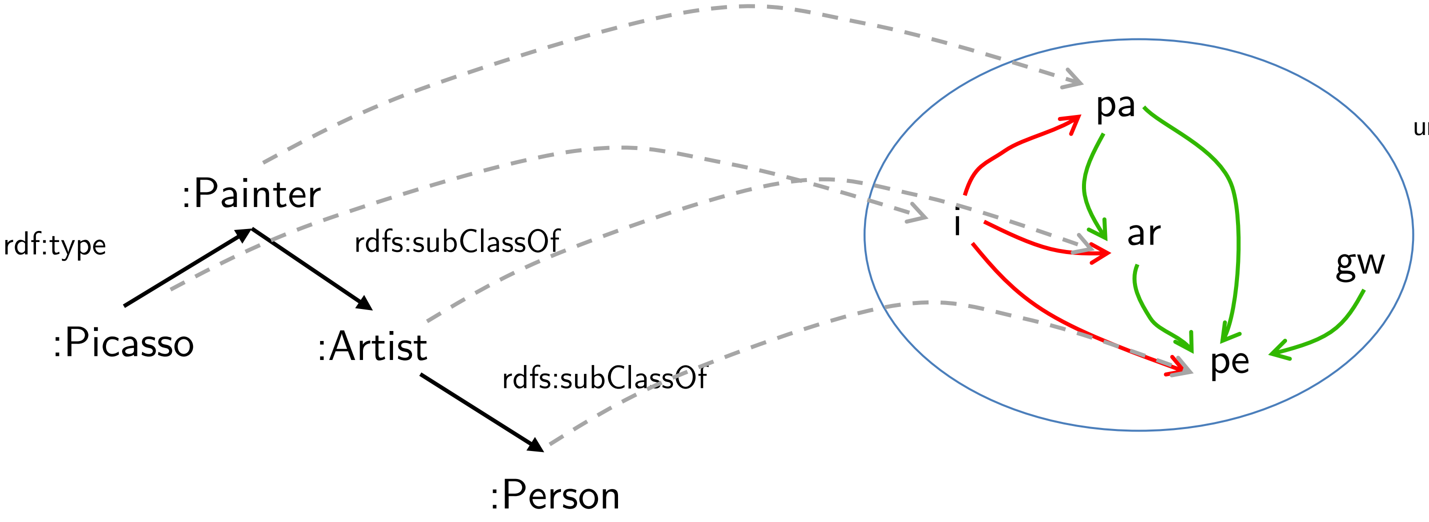
```
# OWL
:VegPizza rdf:type owl:Class ;
    owl:equivalentClass [ rdf:type
                   owl:Restriction ;
                   owl:onProperty :hasTopping ;
                   owl:allValuesFrom :VegTopping
                   ]
```

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



un graph URI fait le lien entre la représenta

rdf:type

rdfs:subClassOf

un graph a une quantité infinie d interprétations possibles

A simple interpretation I is a structure consisting of:

an **interpretation domain** IR (set of resources) → Drote

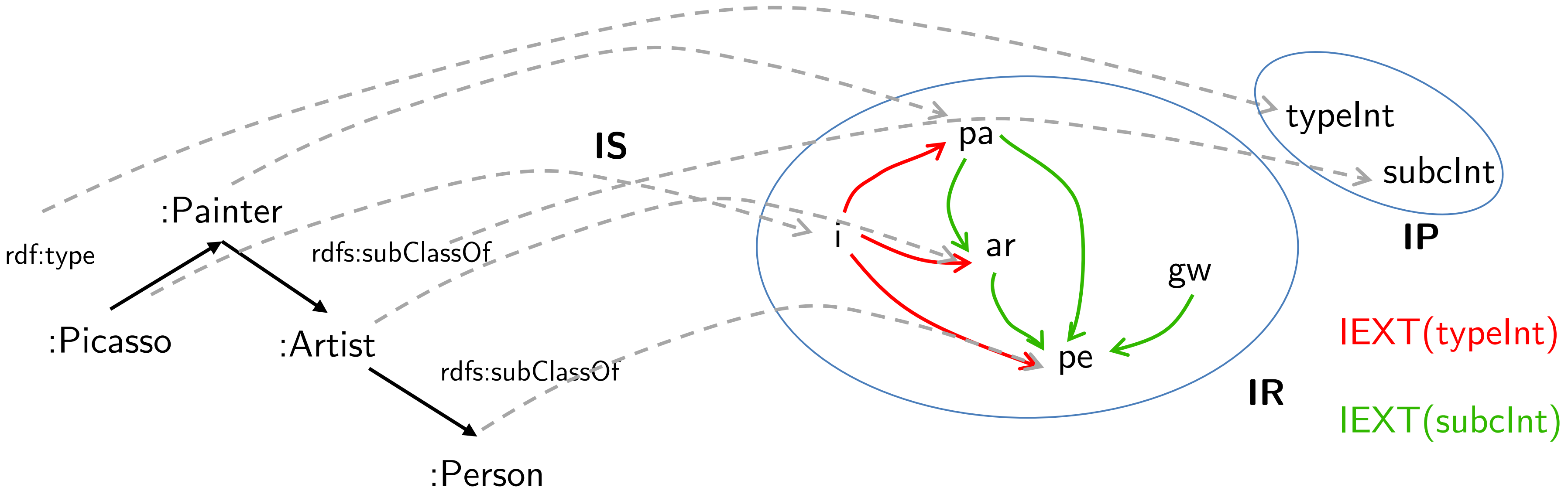
a **set of properties** IP

an **extension mapping** IEXT
associates a binary relation over IR to each p in IP

an **IRI interpretation mapping** IS from IRIs to IR union IP

a **litterals mapping** IL from typed literals to IR

RDF Interpretations



Interpretation (denotation) I of a ground graph (no blank nodes)

- if E is a **typed literal** then $I(E) = IL(E)$
  - non typed literals are interpreted as the string itself
- if E is an **IRI** then $I(E) = IS(E)$

- the interpretation of a **triple** s p o is a value in {true, false}

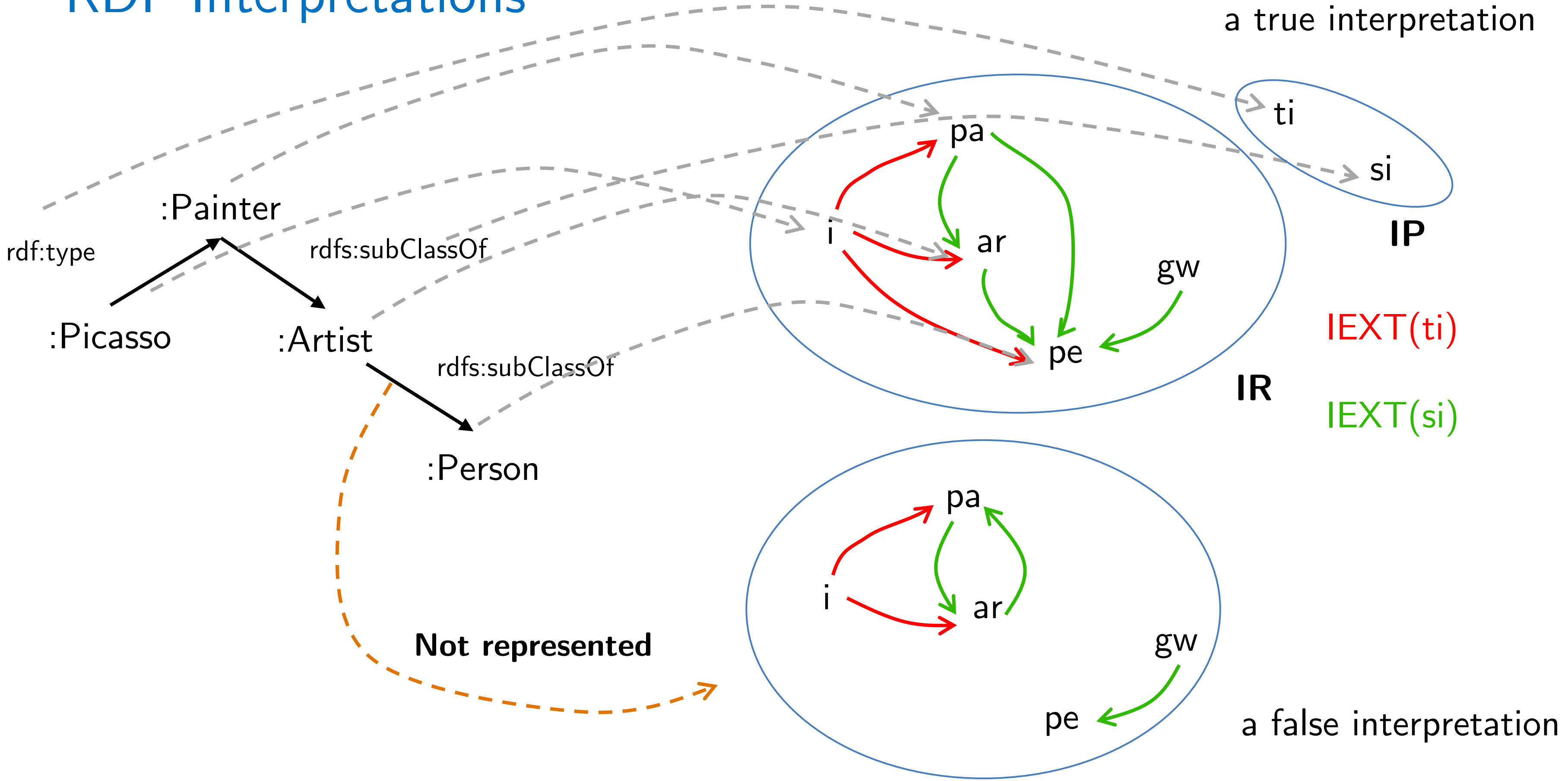
  $I(s\ p\ o)$ = **true** if

  - $I(p)$ is in IP
  - $(I(s), I(o))$ is in $IEXT(I(p))$

  otherwise $I(s\ p\ o)$ = **false**.

- if E is a ground RDF graph then $I(E) = \text{false}$ if $I(E') = \text{false}$ for some triple E' in E, otherwise $I(E) = \text{true}$.

RDF Interpretations



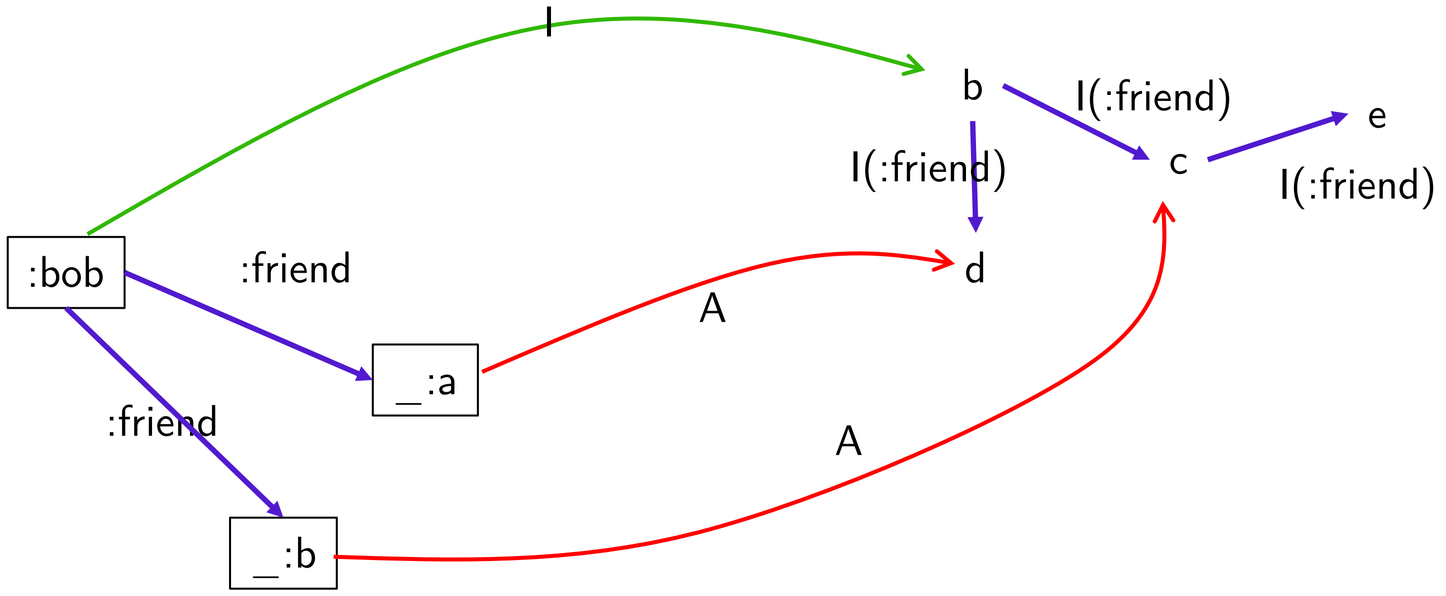
Graphs with blank nodes

Semantic condition for a graph E with blank nodes

- I(E) = true if
  - there is a mapping A from the blank nodes of E to IR
  - I augmented with A is a true interpretation of E

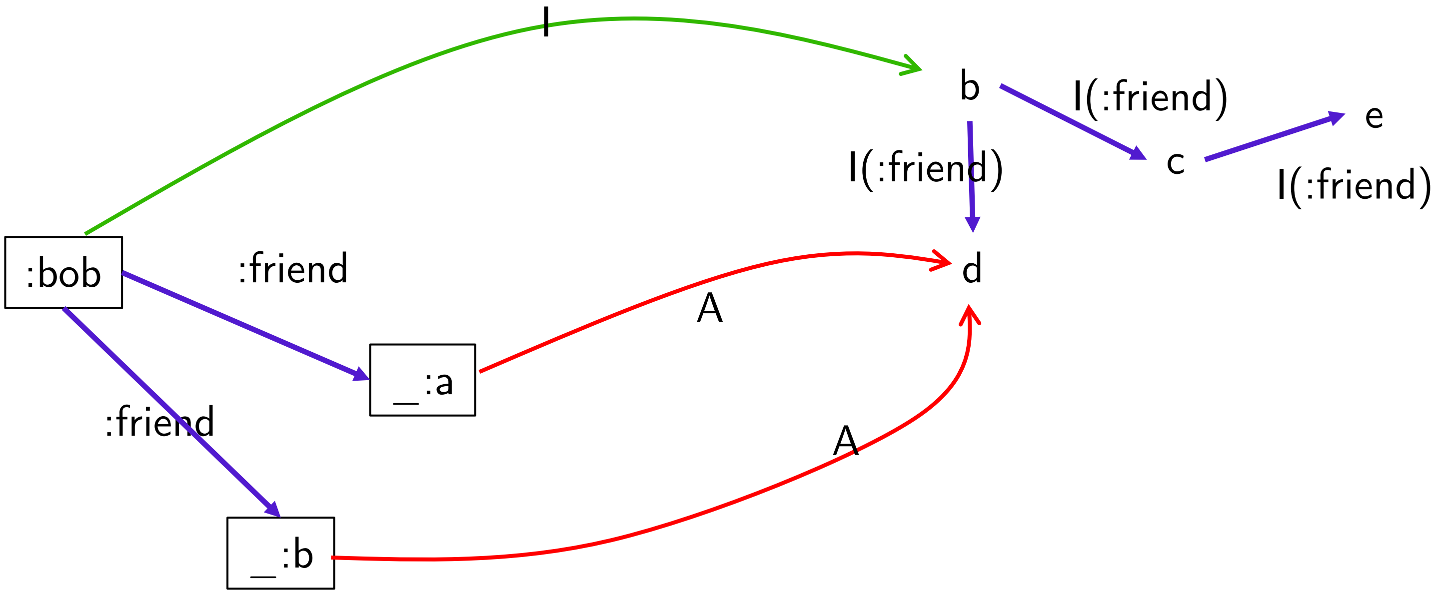
- otherwise I(E)= false.

A true interpretation



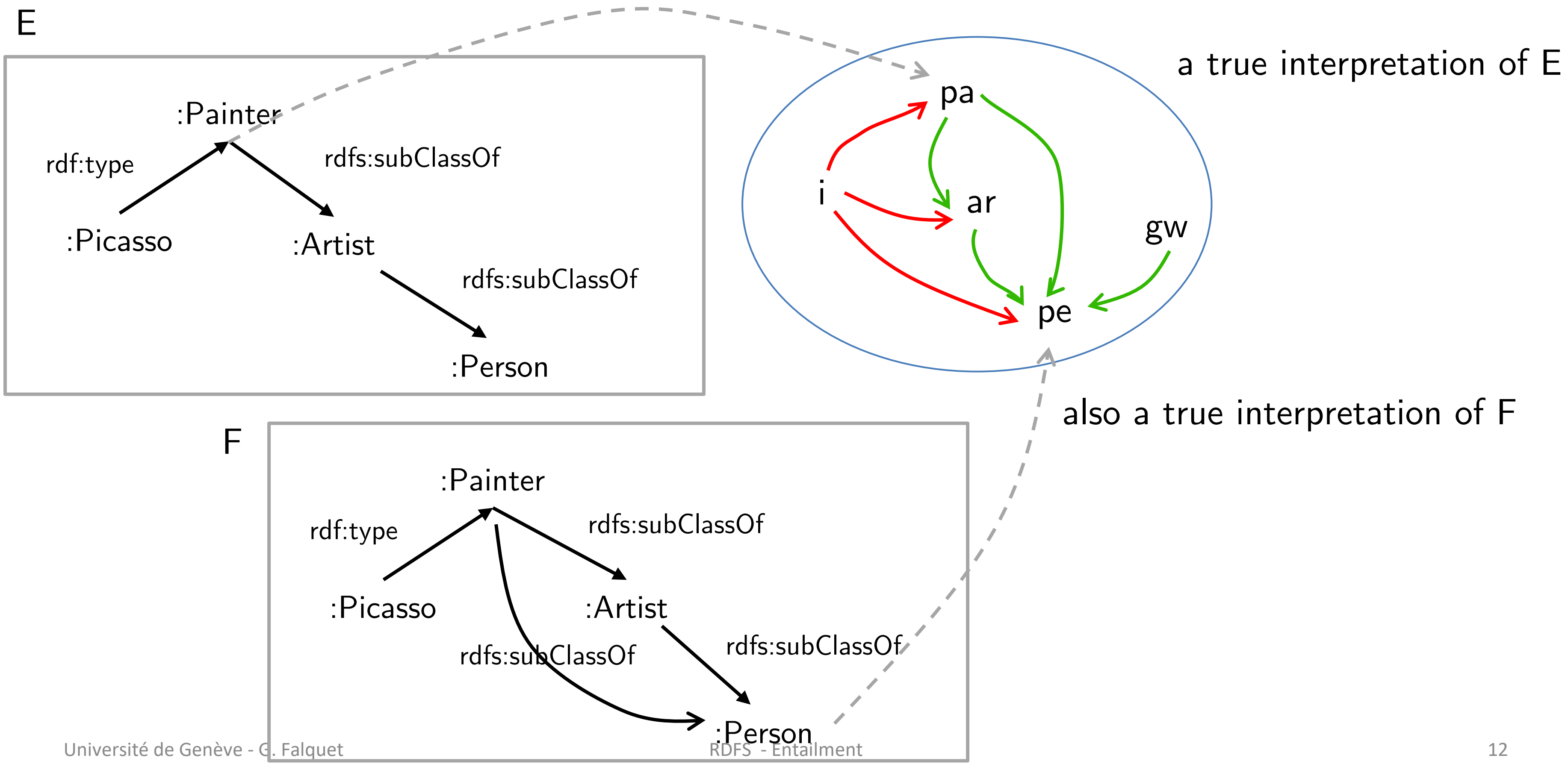
on représente les noeuds blancs

Another choice for A



en RDF on a de la peine à dire que bob a 2 amis, meme avc des URI's, il peuvent pc

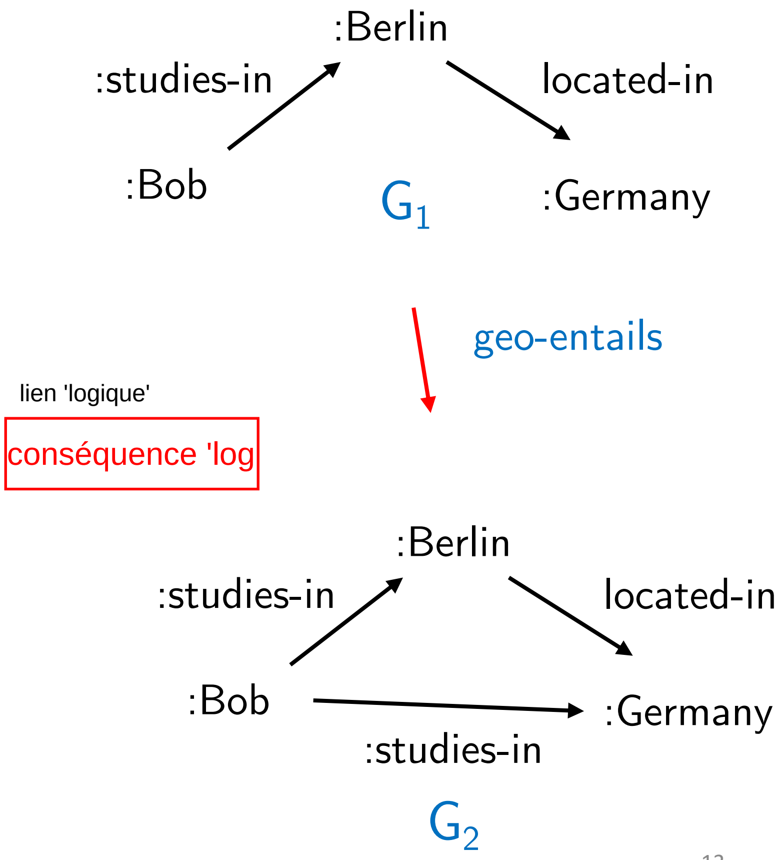
A graph may have several true interpretation



General Notion of Entailment

A relation X between RDF graphs

Represents the notion of logical consequence



X-Entailment for a graph

y a une conséquence logique si E vrai implique F v

A graph E ***X*-entails** a graph F

iff

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

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

= The usual notion of logical consequence

RDFS Interpretations - definitions

RDFS = RDF + class + domaine/range + subclass + subproperty

- ICEXT(c) := { x : (x,c) is in IEXT(I(rdf:type)) }

extension de class

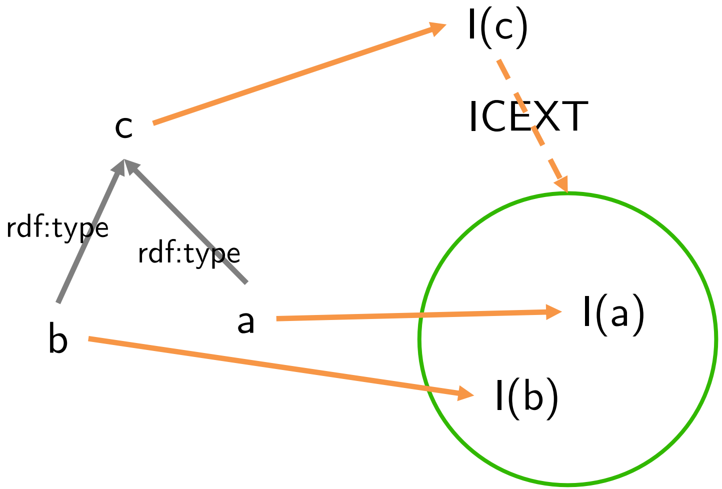
-> regarder slide d'apres pour voir la

- IC := ICEXT(I(rdfs:Class))
- LV := ICEXT(I(rdfs:Literal))
- ICEXT(I(rdfs:Resource)) = IR
- ICEXT(I(rdf:langString)) := {I(E) : E a language-tagged string }

For a set D of datatype IRIs

- for every IRI t in D – {rdf:langString}, ICEXT(I(t)) is the value space of I(t)
  - the range of the lexical-to-value mapping of I(t)
- for every IRI t in D, I(t) is in ICEXT(I(rdfs:Datatype))

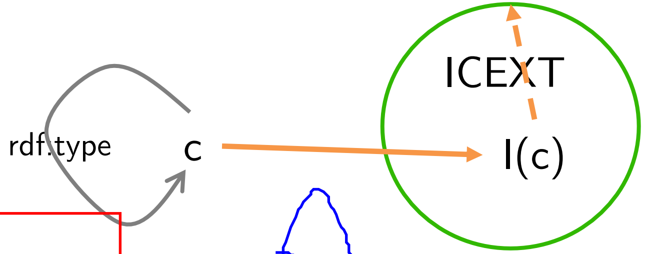
A Theoretical Point



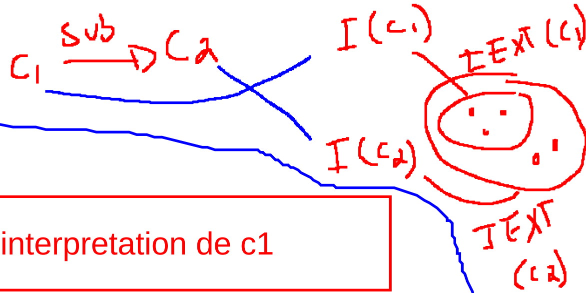
- The interpretation of a class is indirect
  - it is not a subset of IP
  - it is an element of IP that has an extension
- Reason: to be consistent with set theory
  - c rdf:type rdfs:Class. c rdf:type c is legal RDF
  - a direct interpretation would lead to
    - I(c) ∈ I(c) ❌ forbidden in set theory

en théorique des ensembles, l'ensemble X ne peut

mais en RDFS on peut le noter ainsi



RDFS Interpretations – semantic conditions



c1 subclass c2

-> interpretation de c2 doit contenir l interpretation de c1

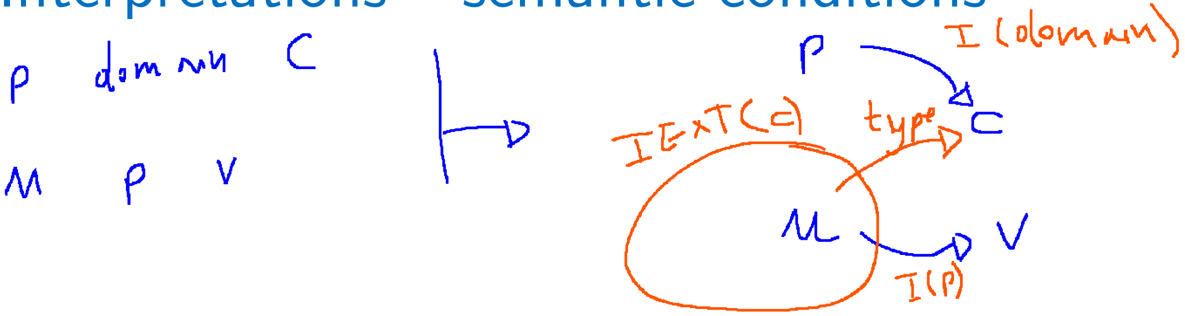
- If $(c_1,c_2) \in \mathrm{IEXT}(\mathrm{I}(\text{rdfs:subClassOf}))$ then c_1 and $c_2 \in \mathrm{IC}$ and $\mathrm{ICEXT}(c1) \subseteq \mathrm{ICEXT}(c_2)$
- IEXT(I(rdfs:subClassOf)) is transitive and reflexive on IC

si c1 est subclass de c2 et c2 subclass c1 -

- If (p, q) is in IEXT(I(rdfs:subPropertyOf)) then p and q $\in$ IP and IEXT(p) $\subseteq$ IEXT(q)
- IEXT(I(rdfs:subPropertyOf)) is transitive and reflexive on IC

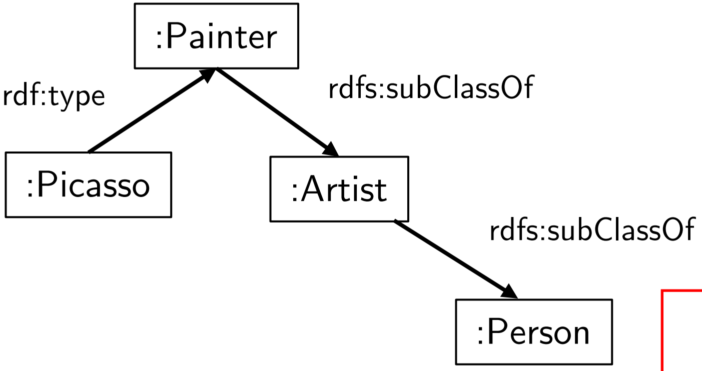
meme chose que avant, mais pour les

RDFS Interpretations – semantic conditions

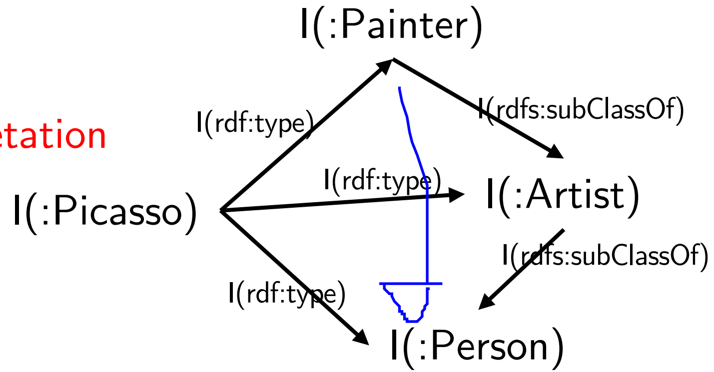


- If (p, c) ∈ IEXT(I(rdfs:domain)) and (u,v) ∈ IEXT(p) then u is in ICEXT(c)

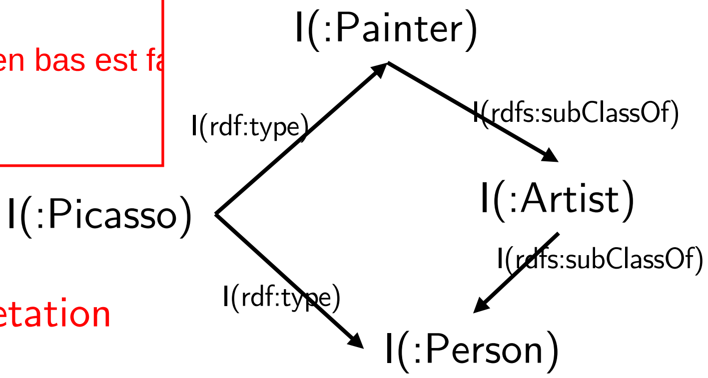
- If (p, c) ∈ IEXT(I(rdfs:range)) and (u,v) ∈ IEXT(p) then v ∈ ICEXT(c) i(sub) -> manquait



true
interpretation



la représentation en bas est fa



false
interpretation

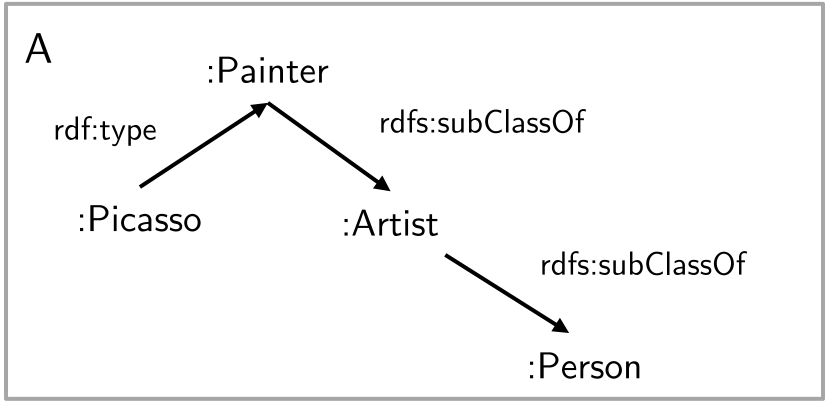
RDFS interpretation – axiomatic triples

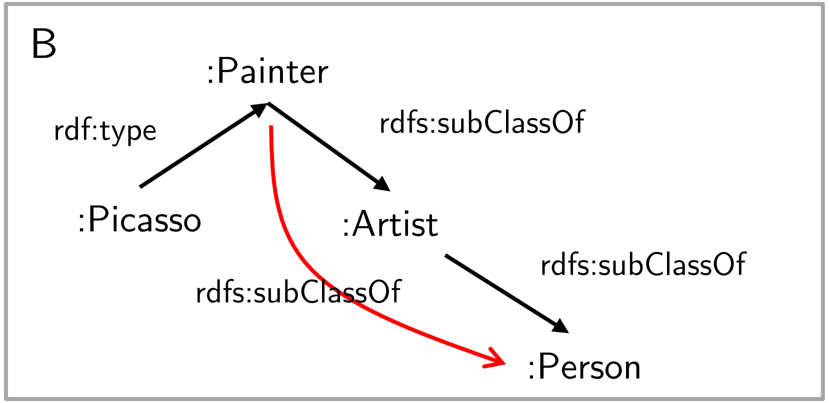
- An RDFS interpretation must satisfy a set of axiomatic triples
  - Each axiomatic triple must have a true interpretation

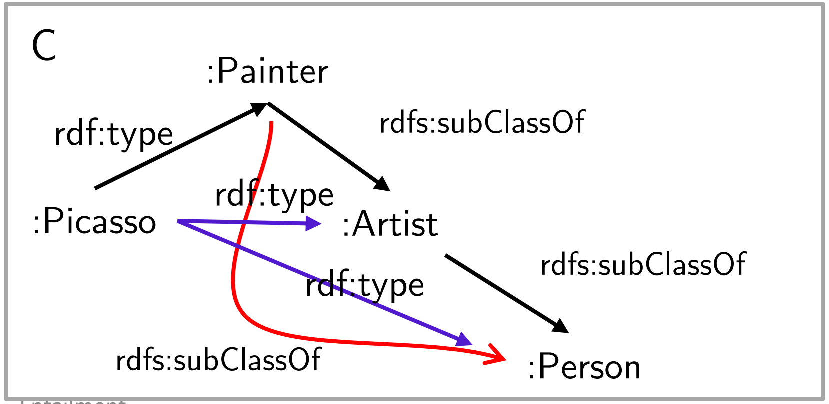
Some axiomatic triples

```
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 .
```

RDFS-Entailments







Any true interpretation of A
is also an true interpretation of B
and a true interpretation of C

Exercises

cours @ 3h00 for example

1. find 3 different true RDF interpretations G with IR = {0, 1, 2, 3, 4}

   G = :a :p :b . :b :q :c

2. find a true RDF interpretation of G that is a false RDFS interpretation of G

   G = :a rdfs:subClassOf :b . :w rdf:type :a .

3. prove that the graph

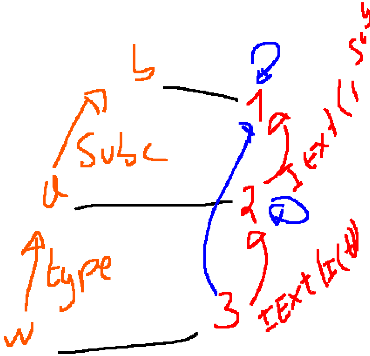
   G = @prefix a : <http://a.a> .

   a rdfs:subClassOf :b . :w rdf:type :a .:w :p :z . :p rdfs:range :a .

   RDFS-entails

   G' = G ∪ :z rdf:type a . :w rdf:type :b . :z rdf:type b .

4. prove that G does not entail G'' = G ∪ :a rdf:type :b



Computing RDFS-Entailment

RDFS entailment can be computed by

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

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

1. :q rdfs:range :d .
2. :p rdfs:subPropertyOf :q .
3. :d rdfs:subClassOf e .
4. :a :p :b

RDFS Entails

5. :a :q :b	by 4. and 2. and rdfs7
6. :b rdf:type :d	by 5. and 1. and rdfs3
7. :b rdf:type :e	by 6. and 3. and rdfs9

The rules are consistent

If G' is inferred by applying the rules to G

then G' is RDFS-entailed by G

The rules are not complete

one ne peut pas tout déduire

deduire*

: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

la on peut tout déduire

deduire*

Semantic web tools that compute entailments

- 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
    - do not automatically add the entailed triples to the graph

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

Entailment and Linked Data

- Other graphs may contain RDF triples that could be used in entailments
- There is not "import" statement in RDF
- There is no automatic import of triples/graphs related to the used vocabularies

⇒ Triples to use in reasoning must be copied into the working graph

My Graph

```
@prefix time: http://www.w3.org/2006/time#.
...
:worldCup19 time:hasBeginning :t1
...
```

```
...
...
```

```
...
time:TemporalEntity a rdfs:Class .
...
time:hasBeginning
    rdfs:domain time:TemporalEntity ;
    rdfs:range time:Instant .
...
```

Time Graph

Summary

- The semantics of RDF is defined by graph interpretations
- Interpretations are used to define the notion of entailment (logical consequence)
- RDFS interpretations are RDF interpretations with additional constraints
- RDFS entailment can be computed by applying inferrence rules
- SW tools perform some form of entailment
  - but they do not import statements about the used vocabulary