Linguaggi Logiche e Tecnologie per la Gestione Semantica dei testi

+

RDFS

Introduction by example

+ RDFS

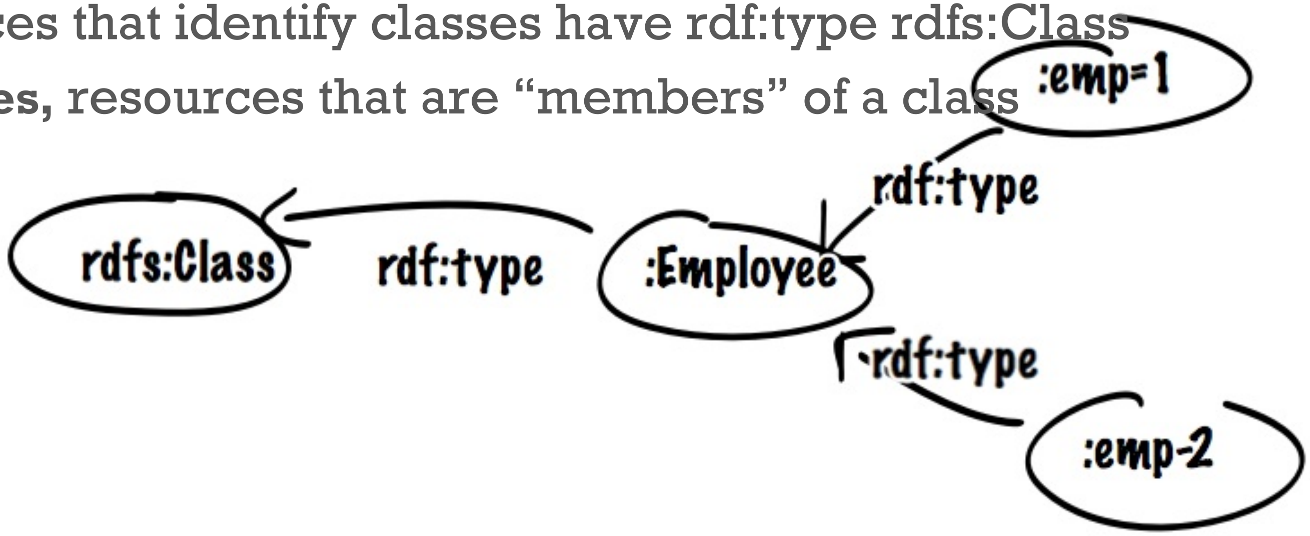
- W3C standard for an ontology language
- RDFS introduces resources (URIs) with a predefined meaning
- Inference engines that support RDFS allow to take that meaning into account
- RDFS inferences **extend** the RDF graph by means of inference and hence, affect **query answering**
- RDFS is very simple compared to **SWRL** or **OWL,** however, it is very useful in many context, allowing for increased productivity, easy data integration and interesting AI applications

Building blocks

- New namespace rdfs: <http://www.w3.org/2000/01/rdf-schema#>

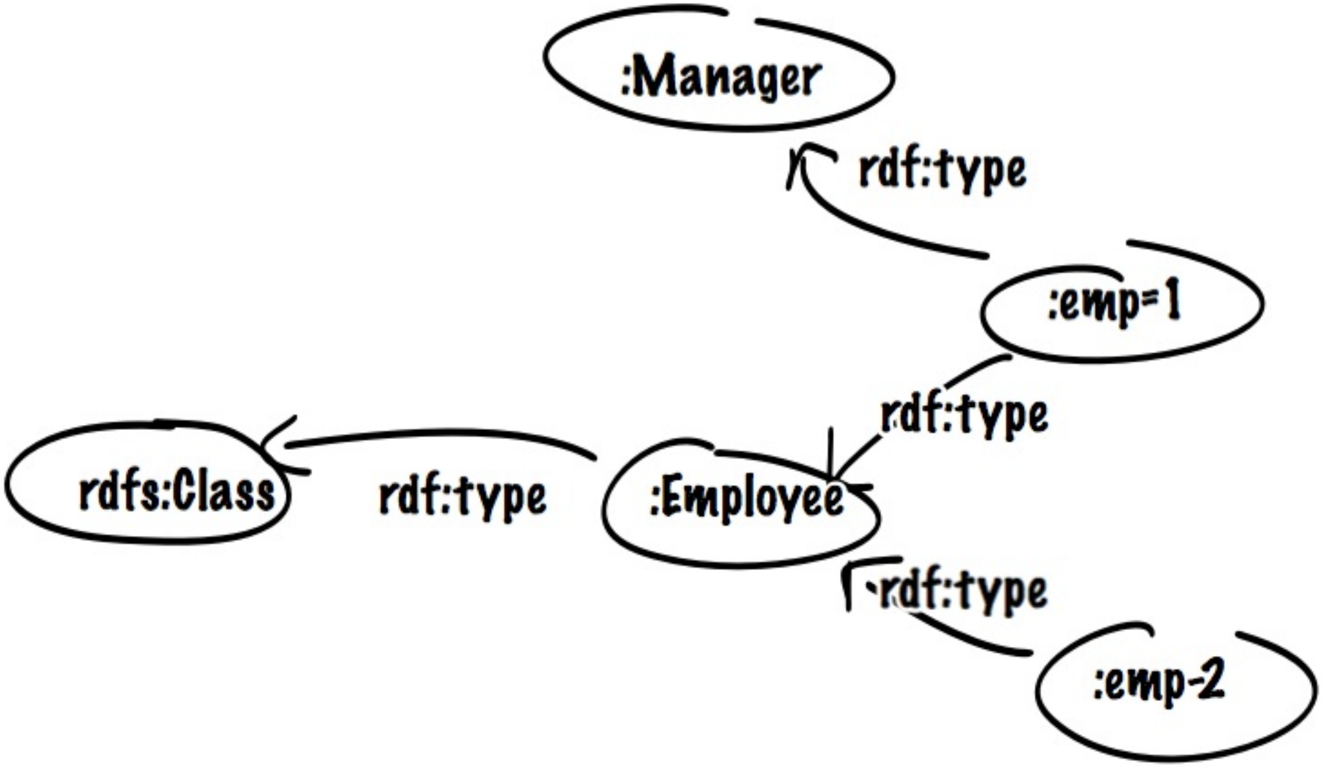
Commonly, Class names are nouns

- New categories:
  - **Classes,** resources that share something in common, allow us to group things together. For example, Employee, Company.
    Resources that identify classes have rdf:type rdfs:Class
  - **Instances,** resources that are "members" of a class



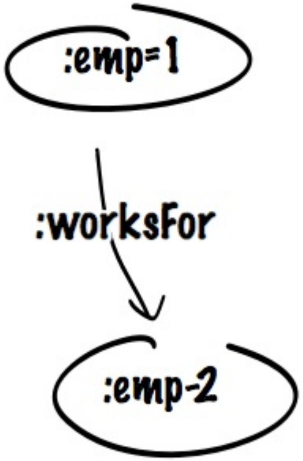
Building blocks

Resources can belong to multiple classes



Building blocks (cont.)

- Properties: Resources used as a predicate in statements



Commonly, Property names are multiple words, expressing direction and in camel-casing

RDFS Ontologies

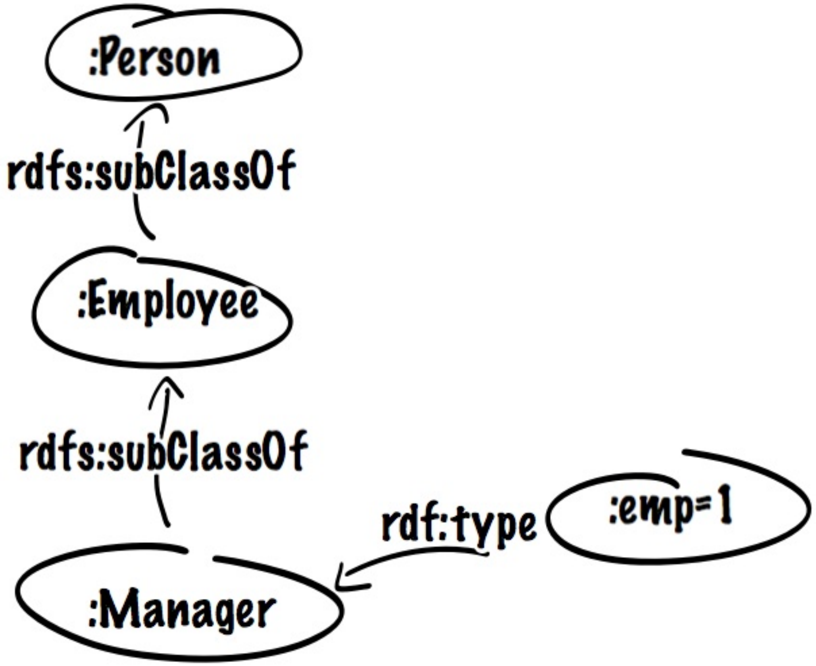
- RDFS Axioms
  - Are RDF triples!

- RDFS ontology is an RDF graph!

- An RDF graph may have a subgraph expressed in RDFS
  - We call the RDFS axioms/triples the Tbox of the ontology (terminological information, predefined meaning)
  - The rest is the Abox of the ontology (plain data, no predefined meaning)

Type propagation

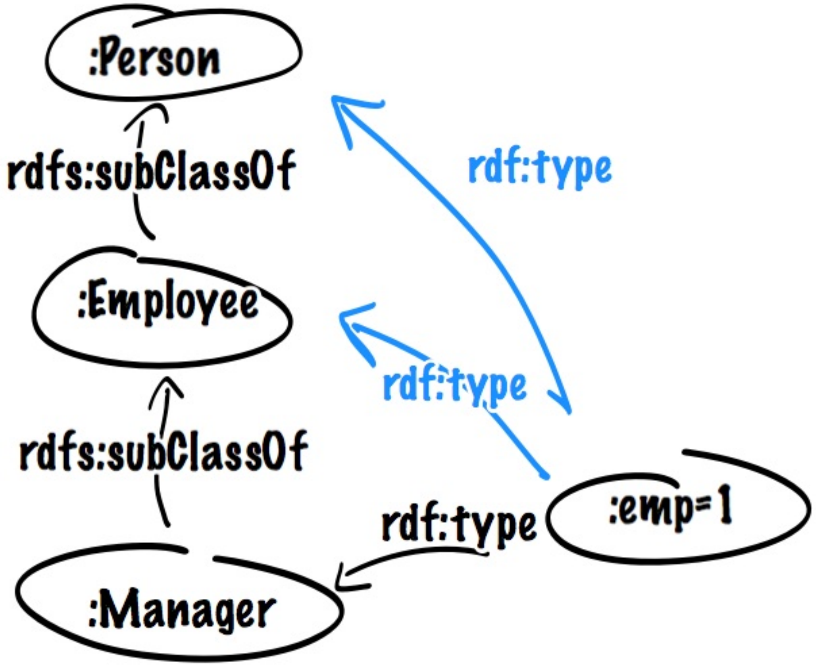
- RDFS vocabulary: rdfs:subClassOf
- Key notions
  - **sub class** (on the left)
  - **super class** (on the right)
- Intuitive meaning, if *:emp=1* is an instance of **subclass** it is also an instance of **superclass**
- Formal meaning: **subsets**
- Inference: **type propagation**



Similar to inheritance in Object Oriented

Type propagation

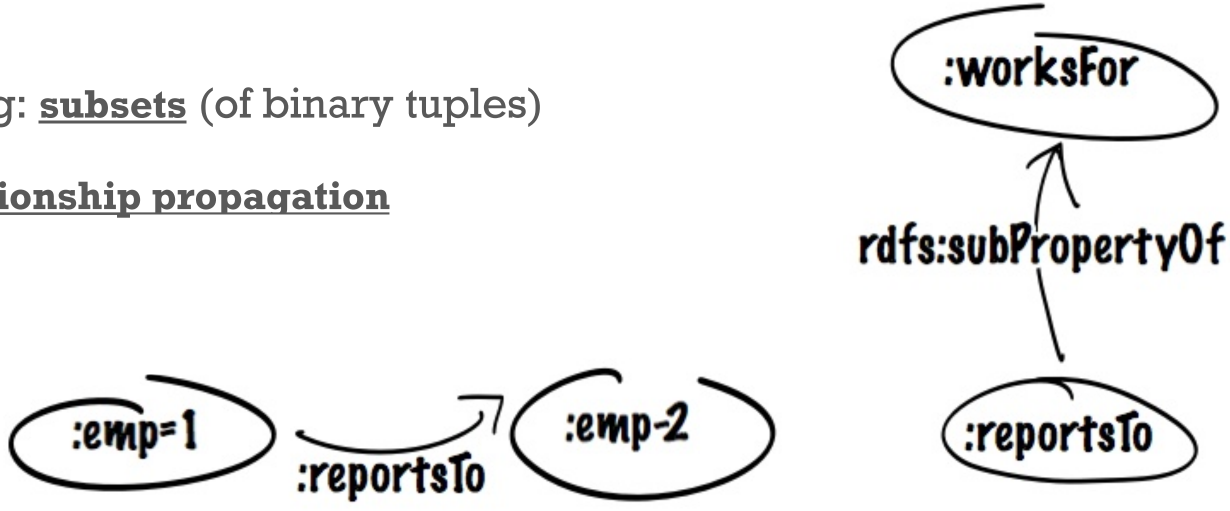
- RDFS vocabulary: rdfs:subClassOf
- Key notions
  - **sub class** (on the left)
  - **super class** (on the right)
- Intuitive meaning, if :*emp=1* is an instance of **subclass** it is also an instance of **superclass**
- Formal meaning: **subsets**
- Inference: **type propagation**



Similar to inheritance in Object Oriented

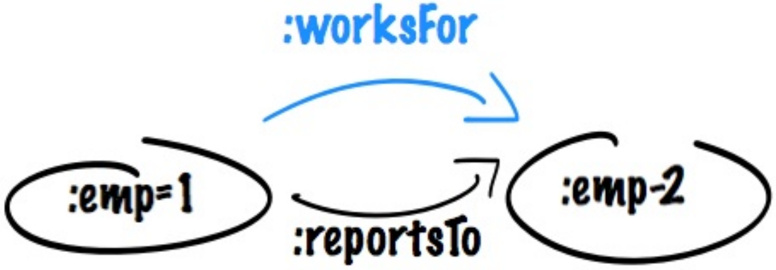
Relation propagation

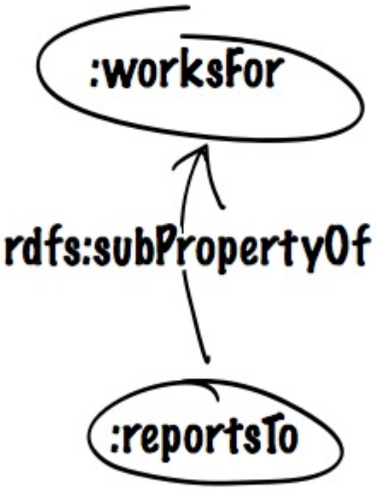
- RDFS vocabulary: rdfs:subPropertyOf
- Key notions
  - **sub property**(on the left)
  - **super property**(on the right)
- Intuitive meaning, if (x,y) are connected with **subproperty** they are also connected with **superproperty**
- Formal meaning: **subsets** (of binary tuples)
- Inference: **relationship propagation**



Relation propagation

- RDFS vocabulary: rdfs:subPropertyOf
- Key notions
  - **sub property**(on the left)
  - **super property**(on the right)
- Intuitive meaning, if (x,y) are connected with **subproperty** they are also connected with **superproperty**
- Formal meaning: **subsets** (of binary tuples)
- Inference: **relationship propagation**





Types by usage

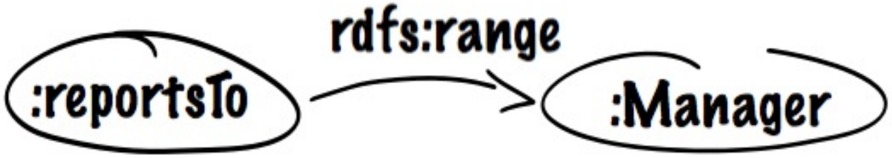
- RDFS vocabulary:
  rdfs:domain, rdfs:range

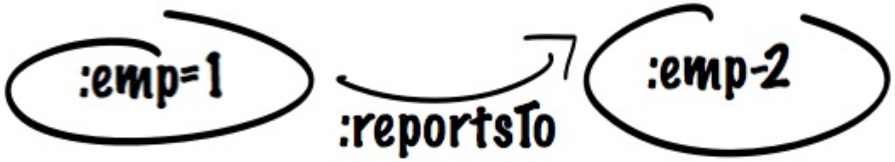
- Key notions
  - **domain of a triple:**
    the subject
  - **range of a triple:**
    the object

- **:p rdfs:domain :C** -> the domain of any triple where :p is the predicate is an instance of :C
  (similar for rdfs:range)

- Formal meaning:
  if (x,y) in P, then x in :C

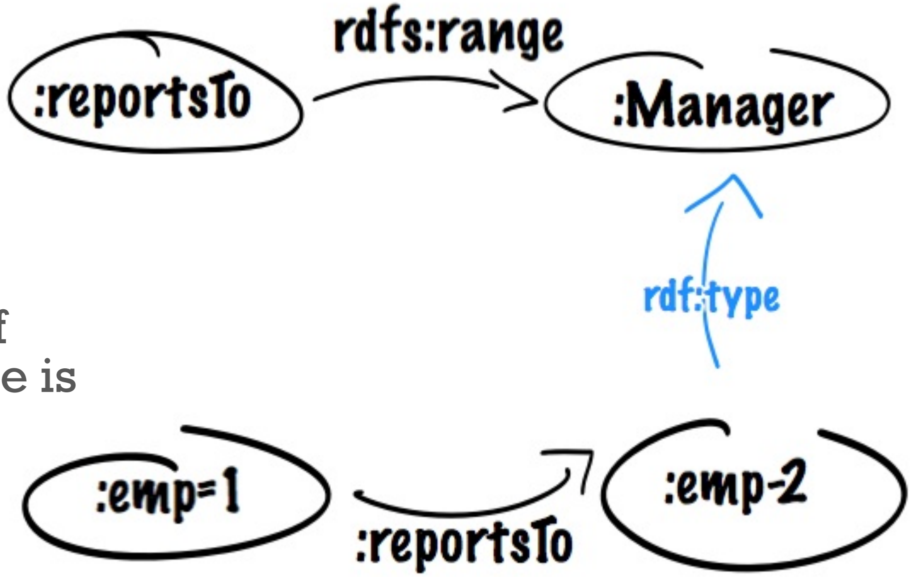
- Inference: type assignment by property usage





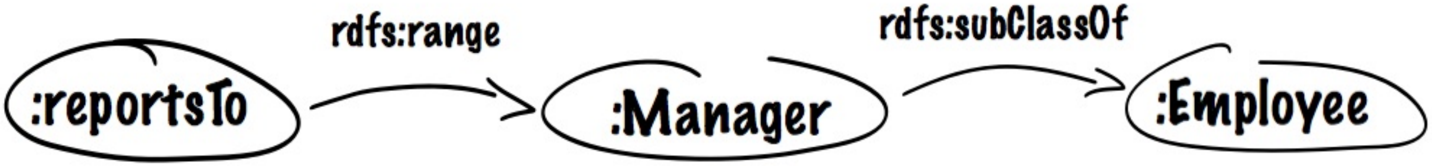
+ Types by usage

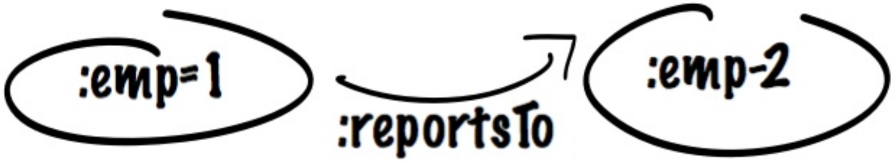
- RDFS vocabulary:
  rdfs:domain, rdfs:range
- Key notions
  - **domain of a triple:**
    the subject
  - **range of a triple:**
    the object
- **:p rdfs:domain :C** -> the domain of any triple where :p is the predicate is an instance of :C
  (similar for rdfs:range)
- Formal meaning:
  Inference: type assignment by property usage



Interactions

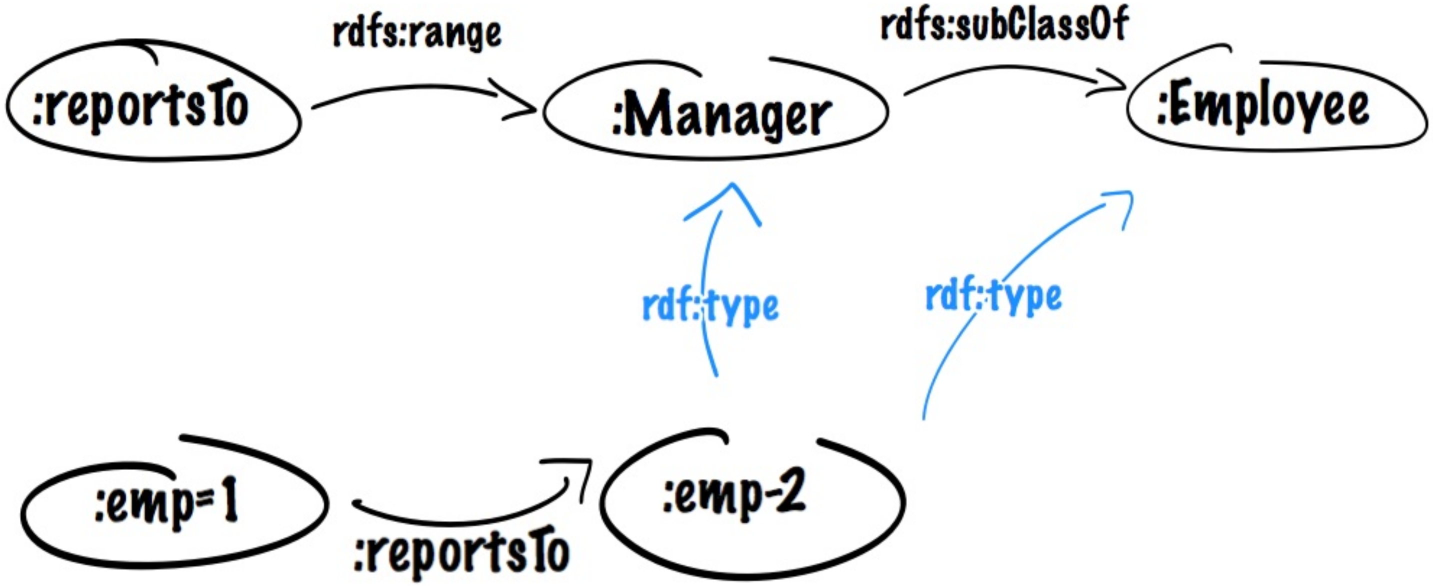
- All inferences interact to allow complex behavior





Interactions

- All inferences interact to allow complex behavior



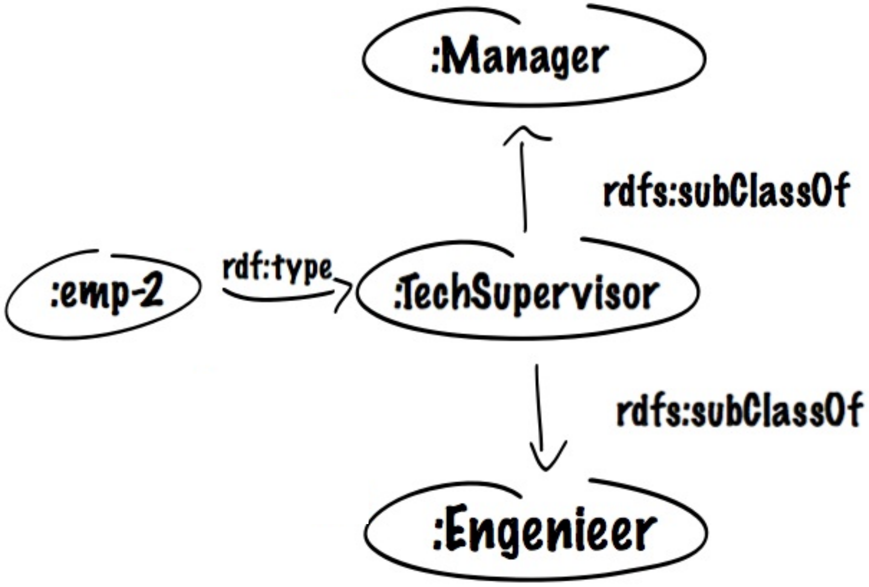
Set intersection

- Proper set intersection is not possible in RDFS

- However, expressing necessary membership to multiple classes is possible, i.e., A subset B AND C

  A rdfs:subClassOf B
  A rdfs:subClassOf C

  consider
  x rdf:type A



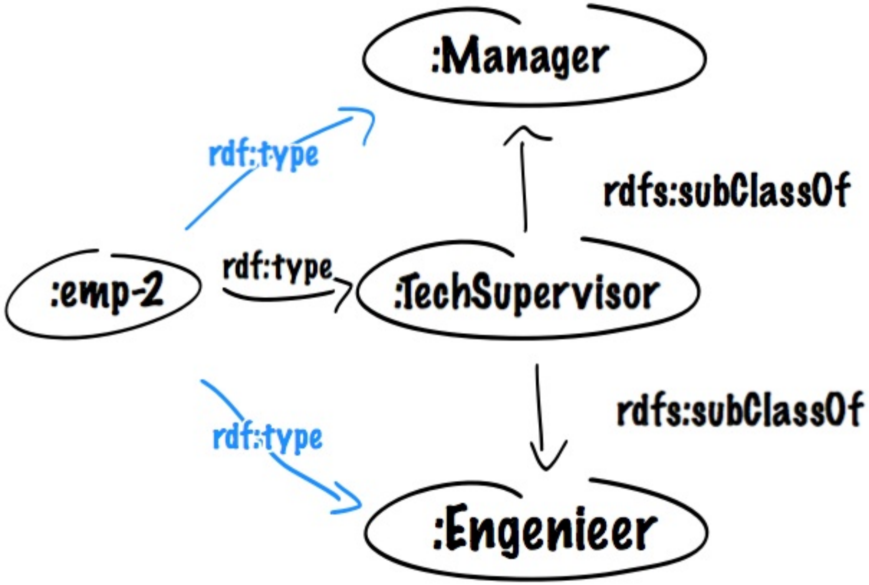
Set intersection

- Proper set intersection is not possible in RDFS

- However, expressing necessary membership to multiple classes is possible, i.e., A subset B AND C

  A rdfs:subClassOf B
  A rdfs:subClassOf C

  consider
  x rdf:type A



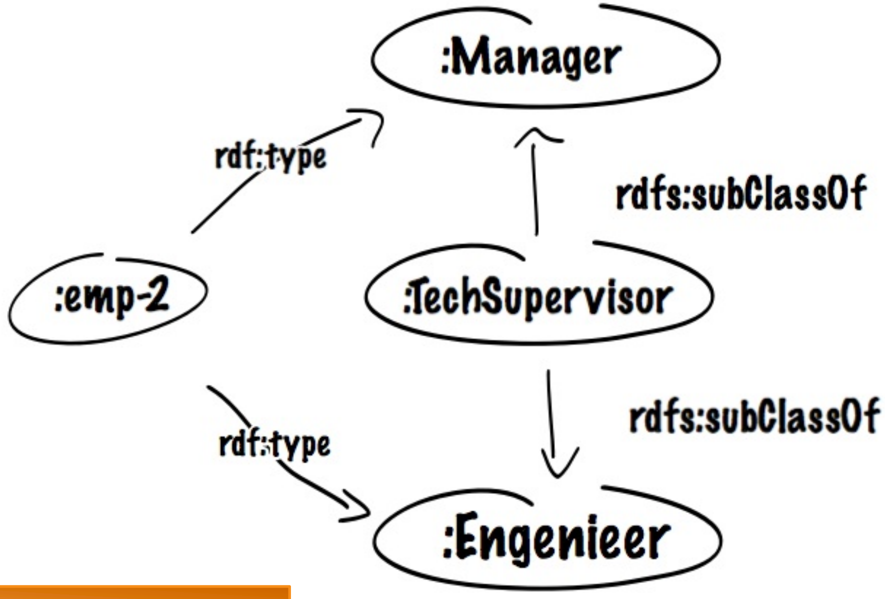
Set intersection

- Proper set intersection is not possible in RDFS

- However, expressing necessary membership to multiple classes is possible, i.e., A subset B AND C

  A rdfs:subClassOf B
  A rdfs:subClassOf C

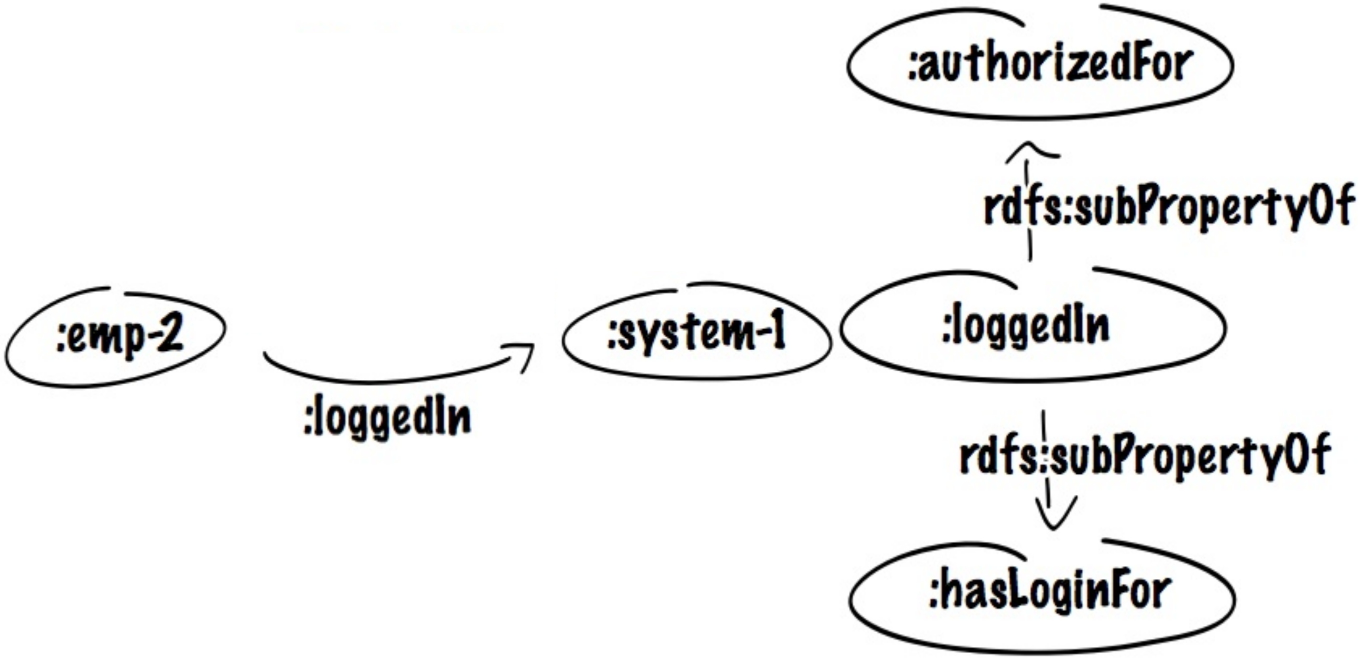
  consider
  x rdf:type A



One direction only!

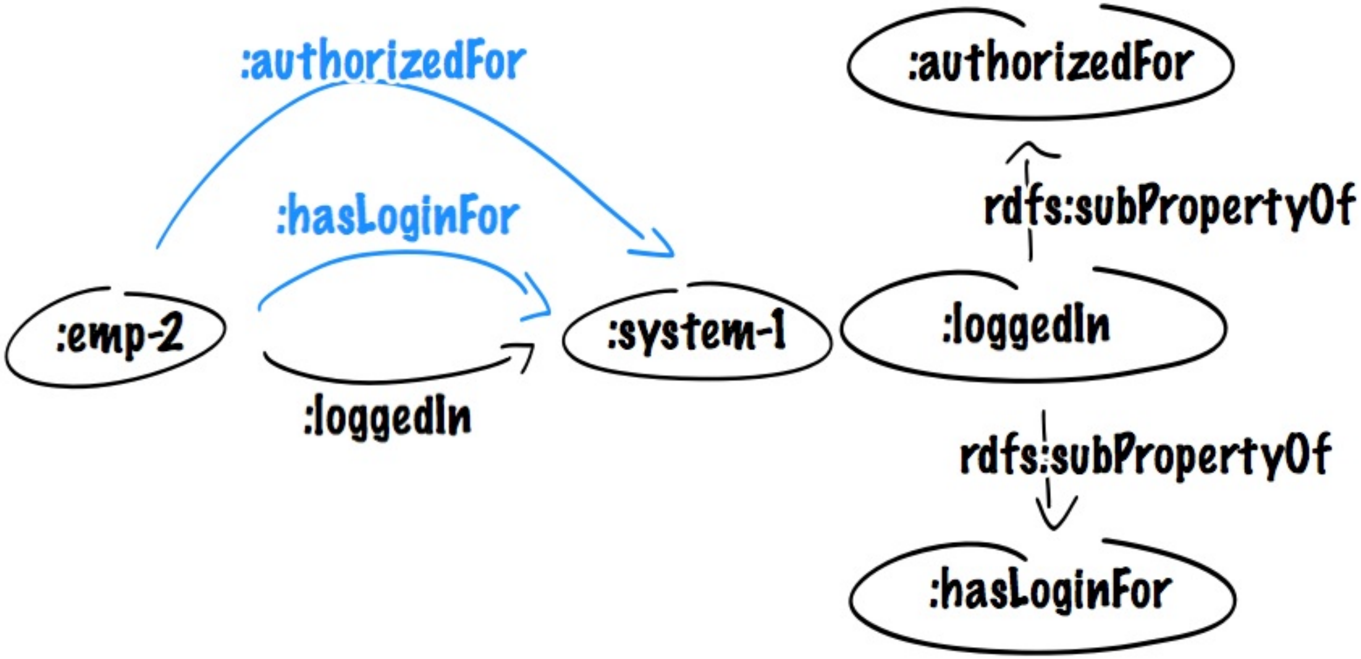
Set intersection

- Similar for roles



Set intersection

- Similar for roles



Set union

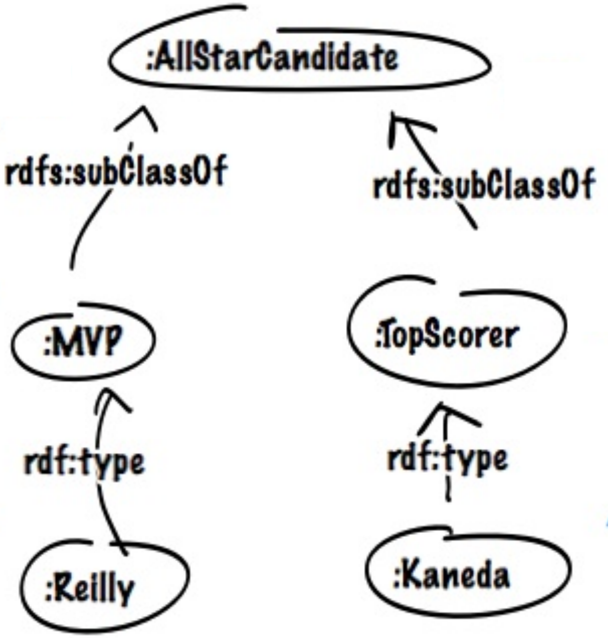
- Proper set union is not possible in RDFS

- However, A OR B subsetOf C

  B rdfs:subClassOf A
  C rdfs:subClassOf A

  consider
  x rdf:type B

  or
  x rdf:type C



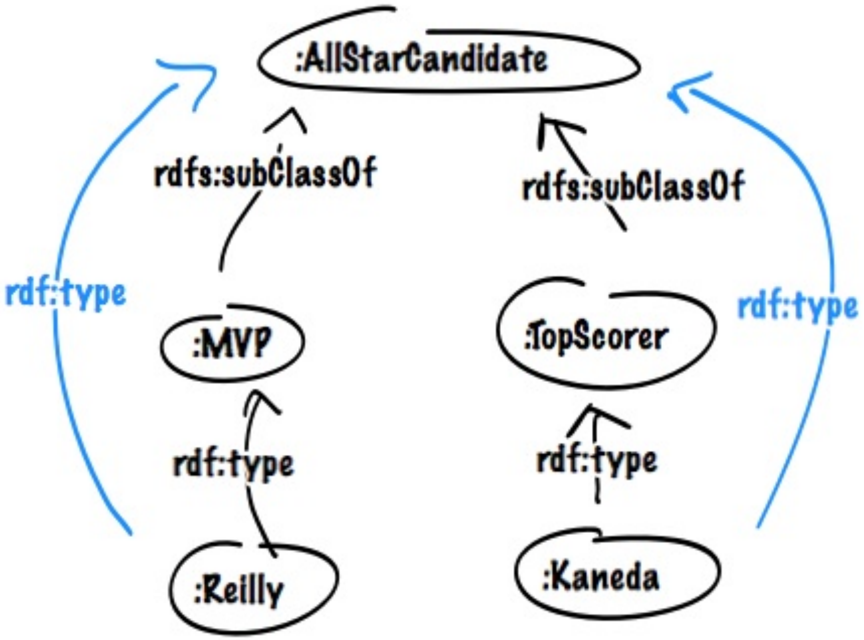
Set union

- Proper set union is not possible in RDFS
- However, A OR B subsetOf C

  B rdfs:subClassOf A
  C rdfs:subClassOf A

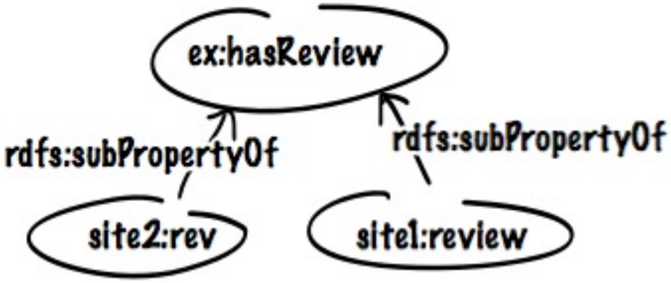
  consider
  x rdf:type B

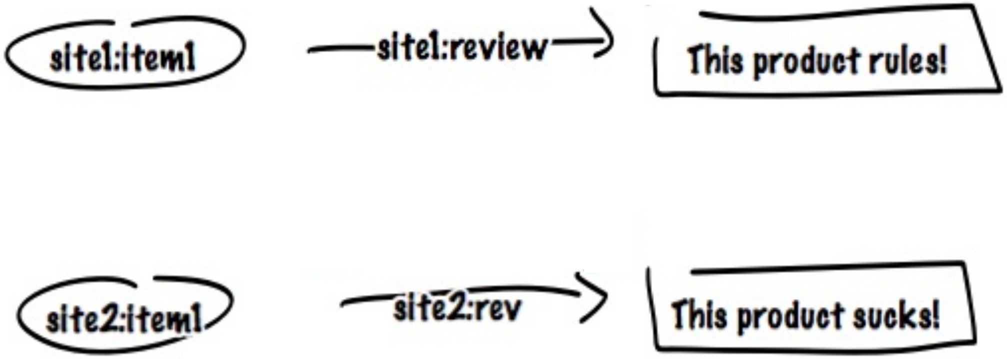
  or
  x rdf:type C



Set union

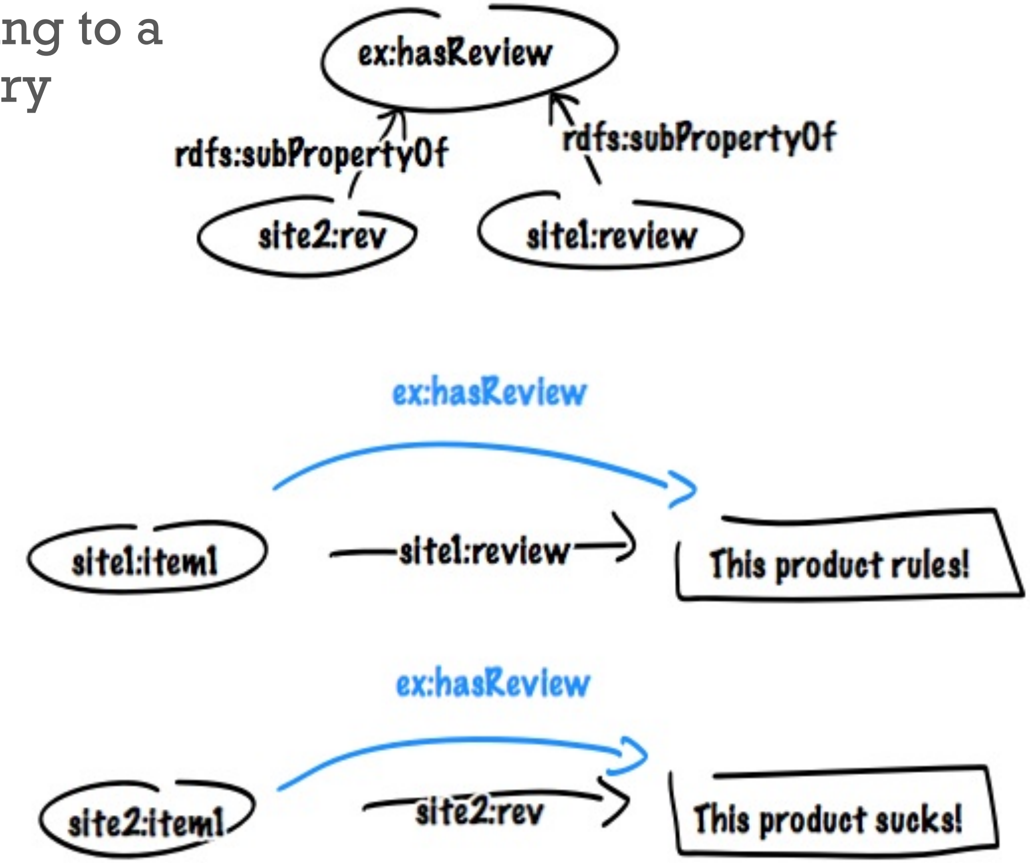
- For roles. Aligning to a global vocabulary





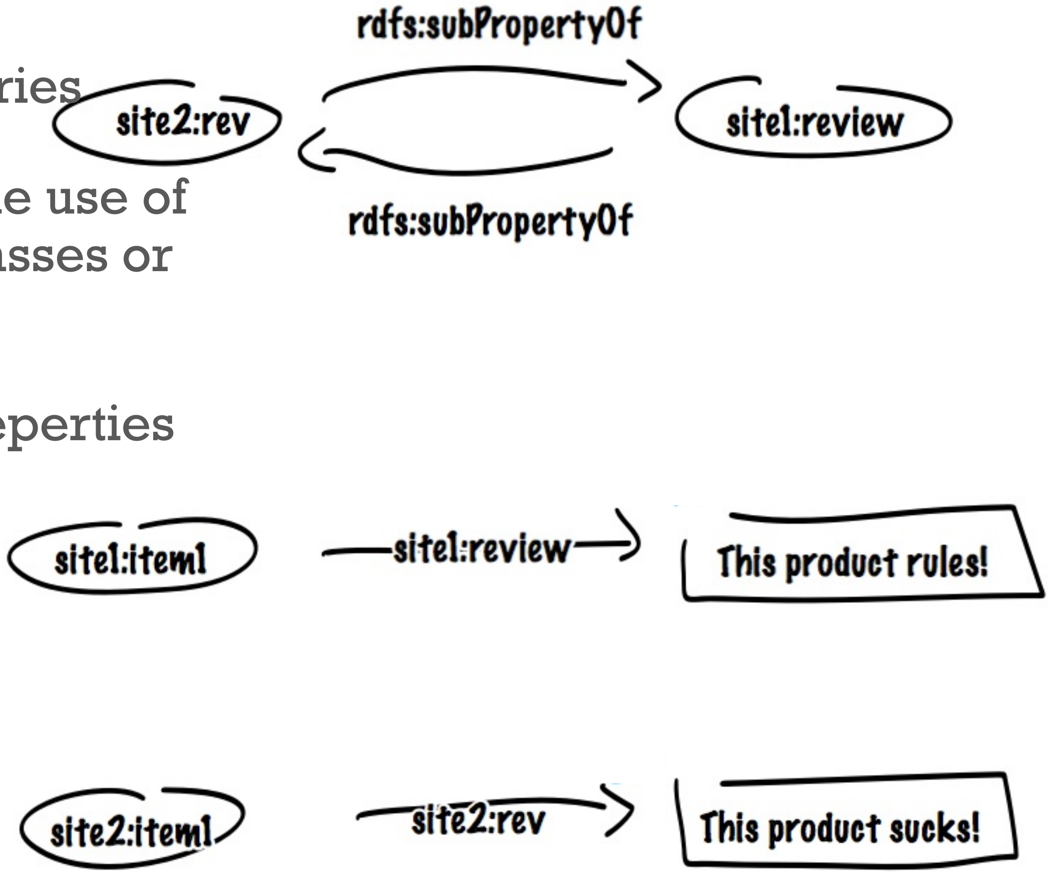
Set union

- For roles. Aligning to a global vocabulary



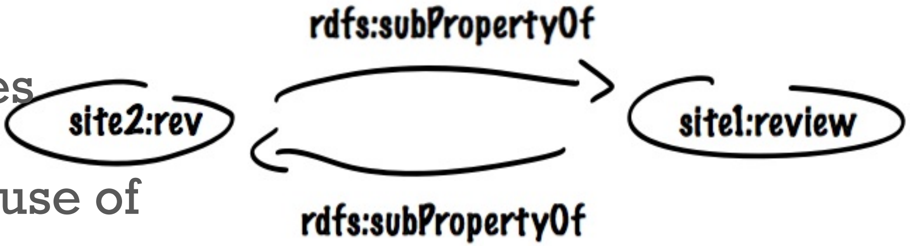
Equivalence

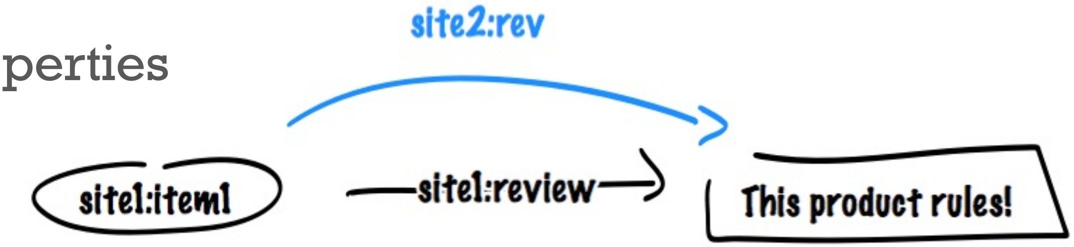
- Merging vocabularies
- To account for same use of different terms (classes or properties)
- For classes or proeperties

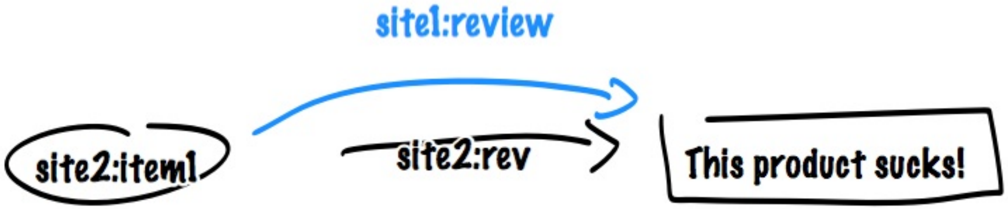


Equivalence

- Merging vocabularies
- To account for same use of different terms (classes or properties)
- For classes or properties


rdfs:subPropertyOf
site2:rev
site1:review
rdfs:subPropertyOf


site2:rev
site1:item1
site1:review
This product rules!


site1:review
site2:item1
site2:rev
This product sucks!