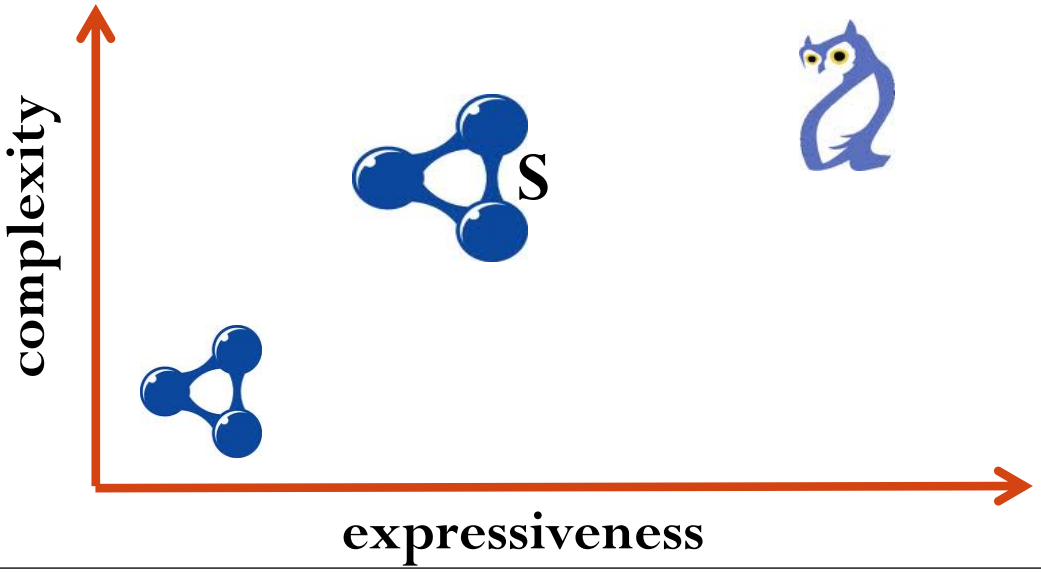
Linguaggi Logiche e Tecnologie per la Gestione Semantica dei testi

RDF+

Ontology languages

- RDF 
- RDF(S) 
- Ontology Web Language (OWL) 
  - OWL lite- OWL DL- OWL Full – OWL2



Ontology Languages

- The expressiveness/complexity depends on the "tools" a language features

- RDF

- RDF (S)

- OWL

RDF-plus (RDF+)

- Augmenting RDF(S) with “a few” OWL constructs

Remark

The semantics of a statement is given by the inference that it can be drawn from it

RDF+ `owl:InverseOf`

- In mathematics the inverse of a function f, denote as f^{-1} is the function that satisfies the property:

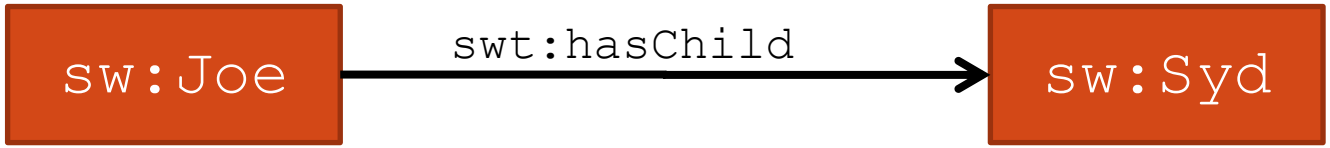
$$f(x) = y \rightarrow f^{-1}(y) = x$$

- In OWL the *inverse of a property* **P** *is another property* **Q**

P `owl:inverseOf` **Q**

$$x \quad P \quad y \rightarrow y \quad Q \quad x$$

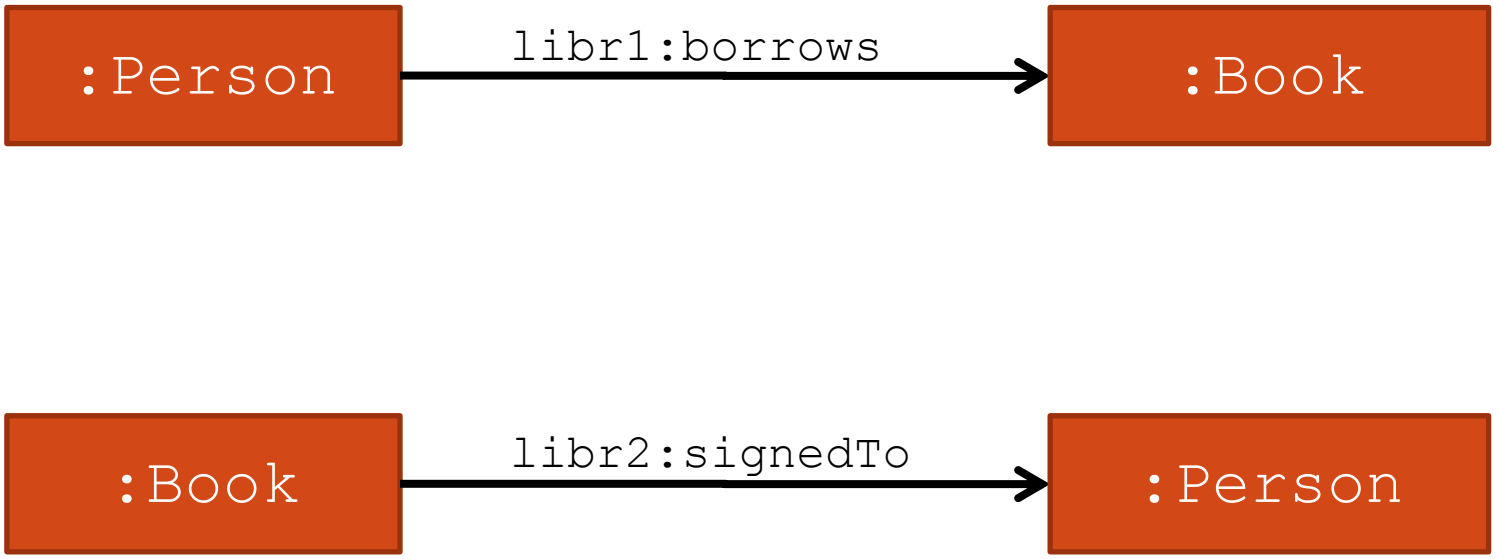
RDF+ : owl:InverseOf



sw:hasChild **owl:inverseOf** sw:hasParent

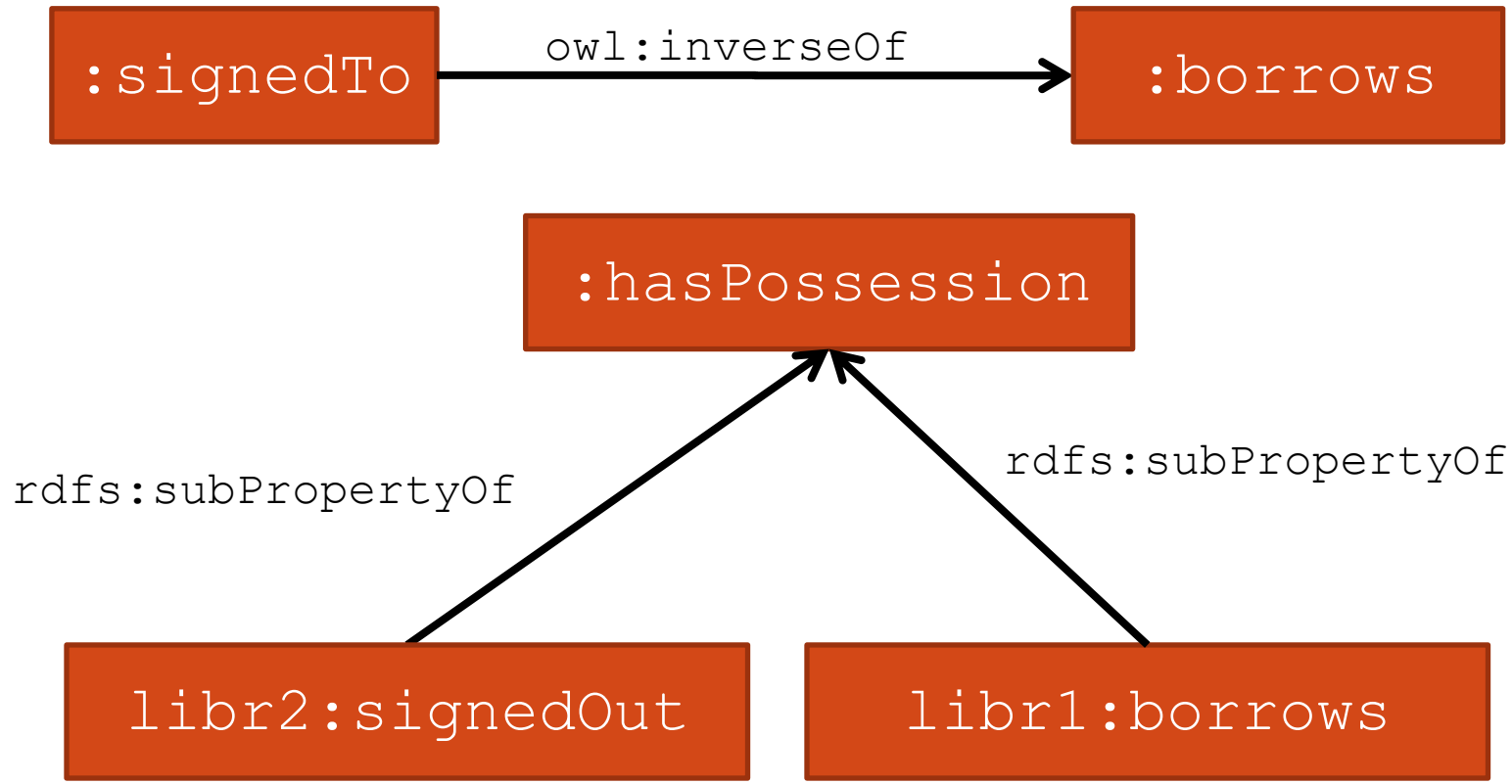
Example - owl:InverseOf

- Suppose to have to merge data where the domain and range of two properties are reversed



Example - owl:InverseOf

- As first step we need to invert domain and range !



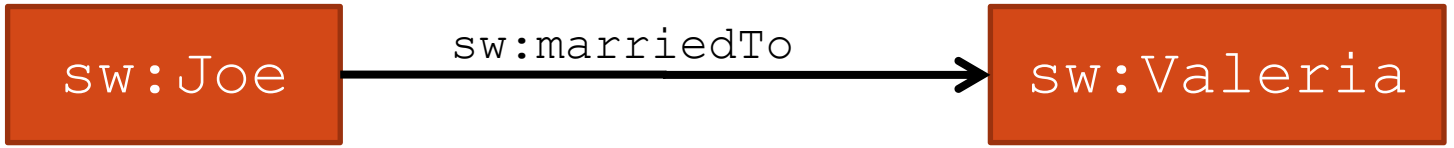
RDF+ `owl:SymmetricProperty`

- Only concerns one property

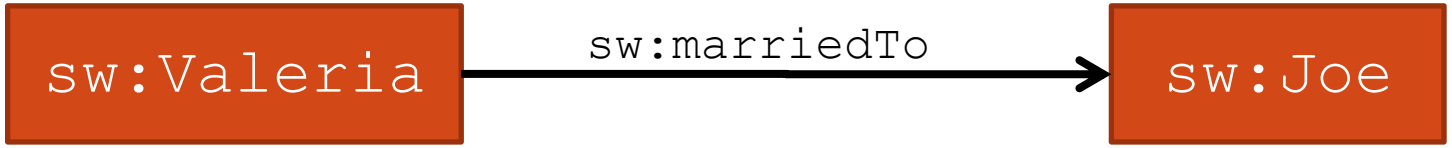
P `rdf:type` `owl:SymmetricProperty`

P `owl:inverseOf` **P**

RDF+ : owl:SymmetricProperty



sw:marriedTo rdf:type **owl:SymmetricProperty**



RDF+ `owl:TransitiveProperty`

- In mathematics a relation R is said to be transitive if:

$$R(a,b), R(b,c) \rightarrow R(a,c)$$

P `rdf:type` `owl:TransitiveProperty`

$$x \quad P \quad y$$

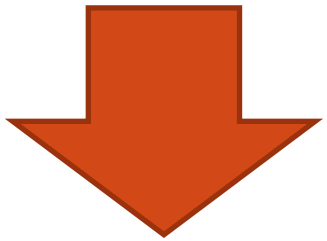
$$y \quad P \quad z$$

$$x \quad P \quad z$$

RDF+ : `owl:TransitiveProperty`

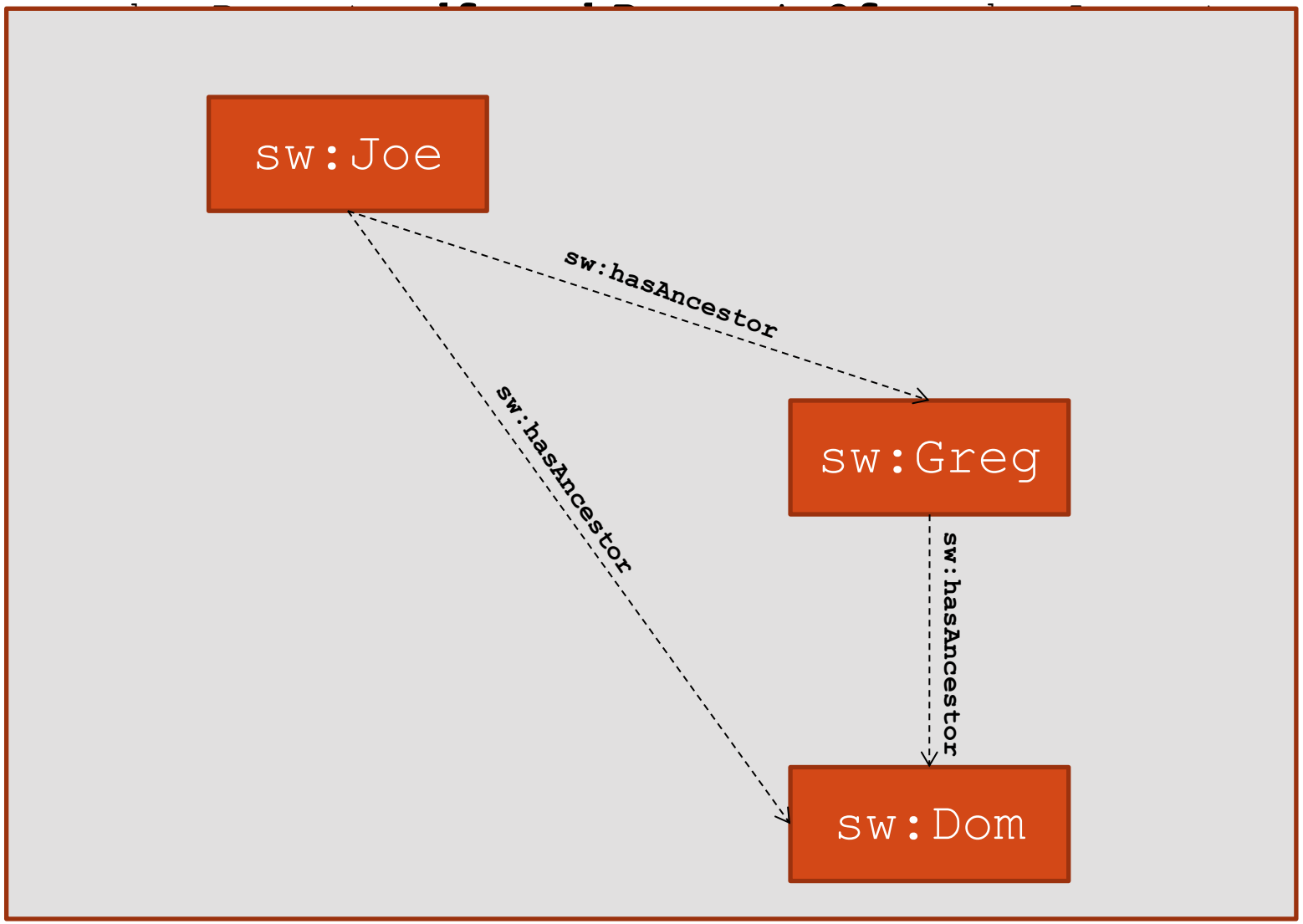
- Relating Parents to Ancestors
- Parents are NOT transistive
  - My parents' parents are NOT my parents
- Ancestors are transistive
  - My parents' ancestors are my ancestors too

How to encode this information ?



using `rdfs:subPropertyOf` and `owl:TransitiveProperty`

RDF+ : owl:TransitiveProperty



RDF+ `owl:equivalentClass`

- When 2 classes have the same members they are defined to be equivalent
- The use of `owl:equivalentClass` does not imply class equality
- Class equality means that the classes have the same intensional meaning (denote the same concept)

RDF+ owl:equivalentClass

A owl:equivalentClass **B**

r rdf:type **A**

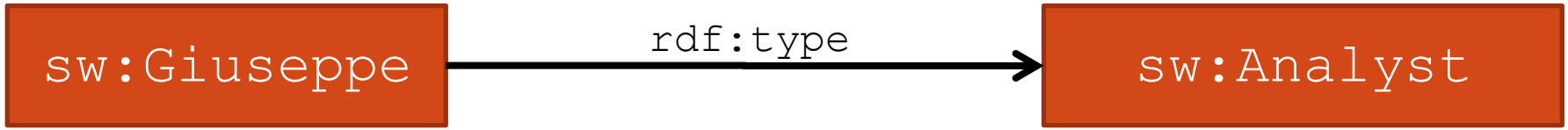
r rdf:type **B**

B rdfs:subClassOf **A**

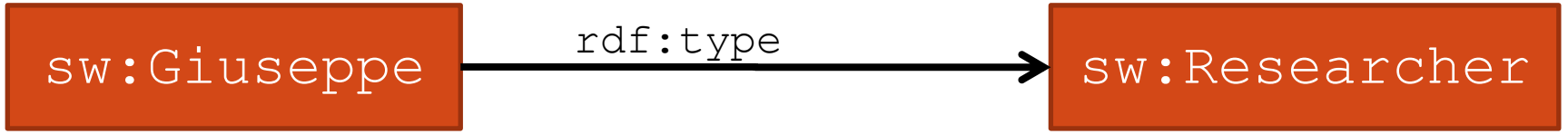
r rdf:type **B**

r rdf:type **A**

RDF+ : owl:equivalentClass



sw:Analyst **owl:equivalentClass** sw:Researcher



RDF+ `owl:equivalentProperty`

- When two properties are equivalent :
  - in any triple that uses one property as a predicate, the other property can be substituted

P `owl:equivalentProperty` **Q**

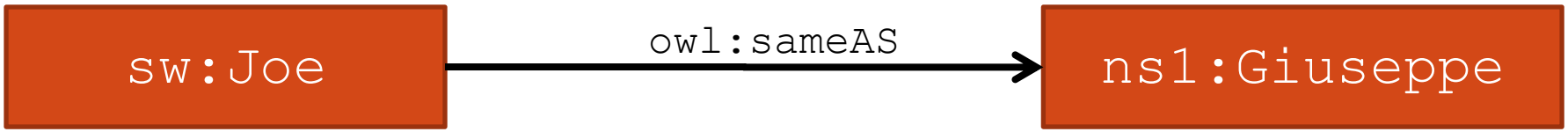
A `P` **B**

A `Q` **B**

RDF+ owl:sameAs

- The construct owl:sameAs links an individual to an individual.
- The statement indicates that two URI references actually refer to the same thing: the individuals have the same "identity"

A owl:sameAs **B**



RDF+ owl:FunctionalProperty

- A functional property can only take one value for any particular individual
- Only one value as object

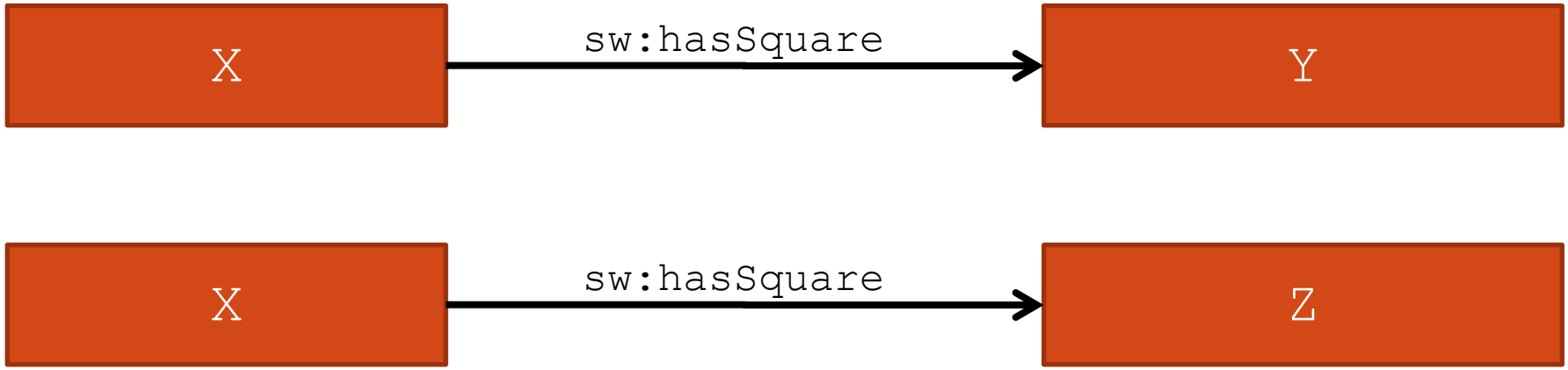
P rdf:type owl:FunctionalProperty

X P **A**

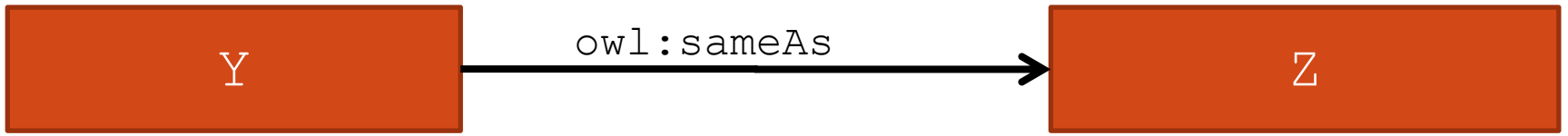
X P **B**

A owl:sameAs **B**

RDF+ owl:FunctionalProperty



sw:hasSquare **owl:FunctionalProperty**



RDF+ owl:InverseFunctionalProperty

- Expresses the opposite of owl:FunctionalProperty
- Only one value as subject

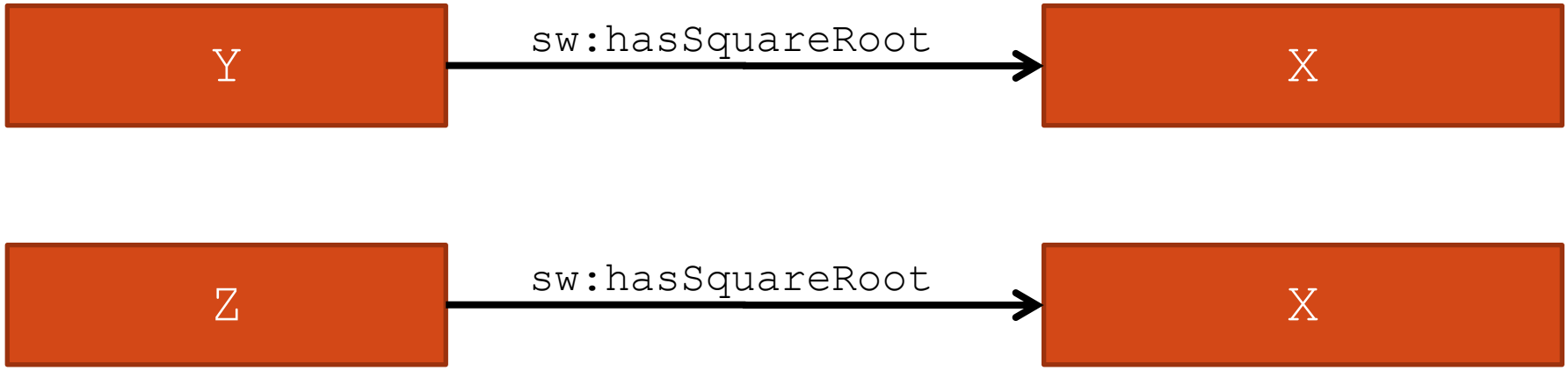
P rdf:type owl:InverseFunctionalProperty

A P **X**

B P **X**

A owl:sameAs **B**

RDF+ owl:InverseFunctionalProperty



sw:hasSquare **owl:InverseFunctionalProperty**