Ontology Languages

Resource Description Framework

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Explain the concept of RDF and RDFS

Objective

Explain the expressive power and limitation of RDF and RDFS

Resource Description Framework (RDF)

What is RDF? (1 of 2)

- The Resource Description Framework (RDF) is a data model for representing information about entities (referred to as “resources”) in the Web
- RDF can be used to represent information about things that can be **identified** on the Web, even when they cannot be directly **retrieved** on the Web

What is RDF? (2 of 2)

- RDF provides a simple graph like data model, with each statement, or triple, representing labeled, directed components.

- Can be identified with a ground atom in FOL

  Predicate(Subject, Object)

- In RDF, subject, predicate, and object names are Internationalized Resource Identifiers (IRIs).

Basic Ideas of RDF

- Resources
- Properties
- Statements

Resources

- **We can think of a resource as an object, a “thing” we want to talk about**
  - authors, books, publishers, places, people, hotels
- **Every resource has an IRI (Internationalized Resource Identifier)**
- **An IRI can be**
  - a URL (Web address) or
  - some other kind of unique identifier
- **Advantages of using IRIs:**
  - A global, worldwide, unique naming scheme
  - Reduces the homonym problem of distributed data representation

Properties

Properties are a special kind of resources

They describe relations between resources

- E.g. “written by”, “age”, “title”, etc.

Properties are also identified by IRIs

Statements

Statements assert the properties of resources

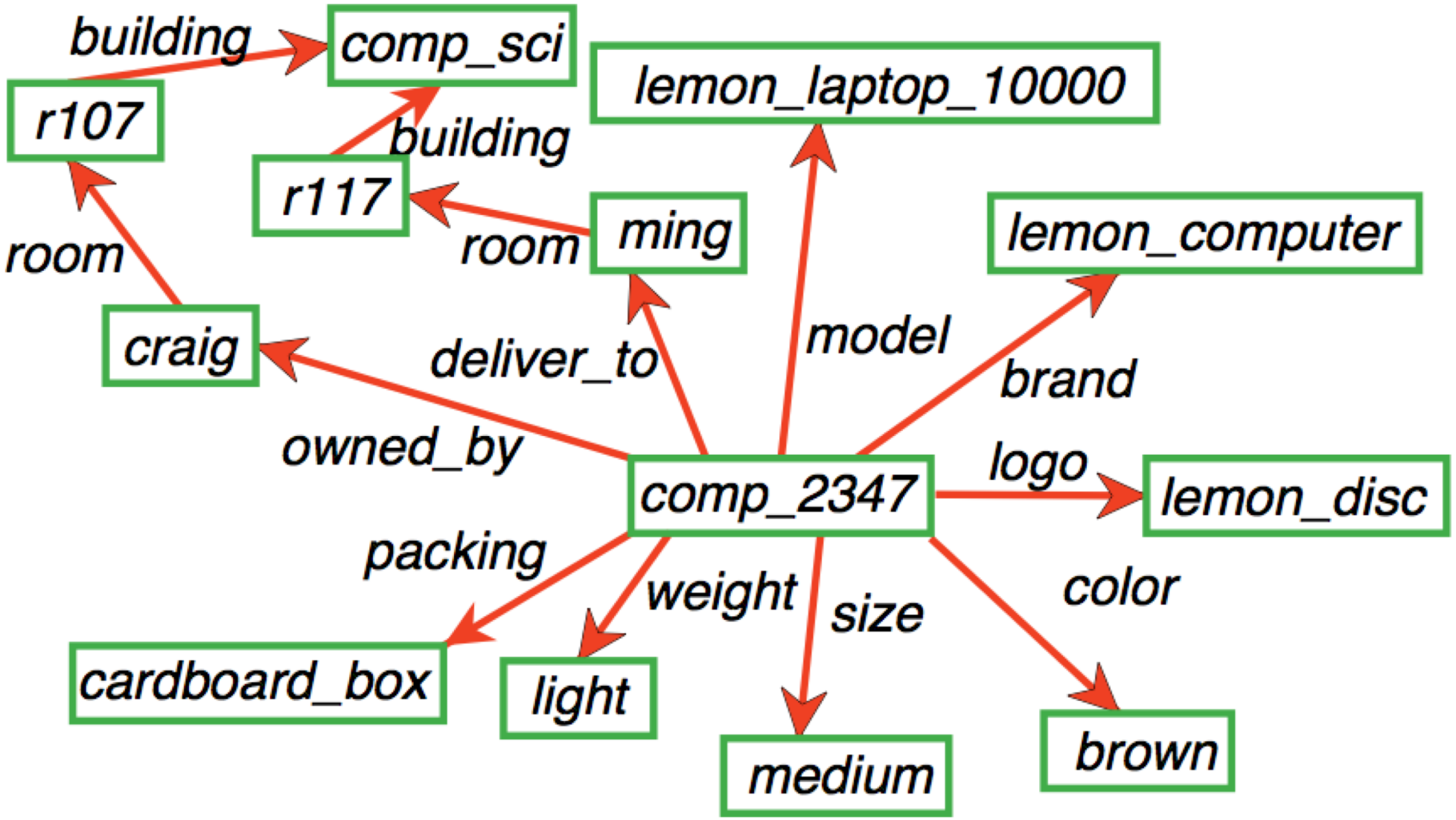
A statement is a subject, predicate, object triple

- It consists of a resource, a property, and a value

Objects can be resources or literals

- Literals are atomic values (strings)

Example


building
comp_sci
lemon_laptop_10000
r107
building
r117
room
ming
lemon_computer
room
craig
deliver_to
model
brand
owned_by
comp_2347
logo
lemon_disc
packing
weight
size
color
cardboard_box
light
medium
brown

Reification

- How to represent *scheduled*(*cse579*, 2, 1030, *byeng210*). "section 2 of course *cse579* is scheduled at 10:30 in room *byeng210*."

- Let *b*123 name the booking:

  prop(*b*123, *course*, *cse579*).
  prop(*b*123, *section*, 2).
  prop(*b*123, *time*, 1030).
  prop(*b*123, *room*, ~~*cse579*~~ byeng210).

- We have **reified** the booking.

- **Reify** means to make into an object.

- **Q:** What if we want to add the year?

  prop(b123, year, 2019)

From Tables to RDF (1 of 2)

Product						
ID	**Model Number**	**Division**	**Product Line**	**Manufacture Location**	**SKU**	**Available**
1	ZX-3	Manufacturing support	Paper machine	Sacramento	FB3524	23
2	ZX-3P	Manufacturing support	Paper machine	Sacramento	KD5243	4
3	ZX-3S	Manufacturing support	Paper machine	Sacramento	IL4028	34
4	B-1430	Control engineering	Feedback line	Elizabeth	KS4520	23
5	B-1430X	Control engineering	Feedback line	Elizabeth	CL5934	14
6	B-1431	Control engineering	Active sensor	Seoul	KK3945	0
7	DBB-12	Accessories	Monitor	Hong Kong	ND5520	100
8	SP-1234	Safety	Safety valve	Cleveland	HI4554	4
9	SPX-1234	Safety	Safety valve	Cleveland	OP5333	14

From Tables to RDF (2 of 2)

Subject	Predicate	Object
mfg:Product1	mfg:Product_ID	1
mfg:Product1	mfg:Product_ModelNo	ZX-3
mfg:Product1	mfg:Product_Division	Manufacturing support
mfg:Product1	mfg:Product_Product_Line	Paper machine
mfg:Product1	mfg:Product_Manufacture_Location	Sacramento
mfg:Product1	mfg:Product_SKU	FB3524
mfg:Product1	mfg:Product_Available	23
…	…	…
mfg:Product1	rdf:type	mfg:Product
mfg:Product2	rdf:type	mfg:Product
…	…	…

RDF Schema

Basic Ideas of RDF Schema

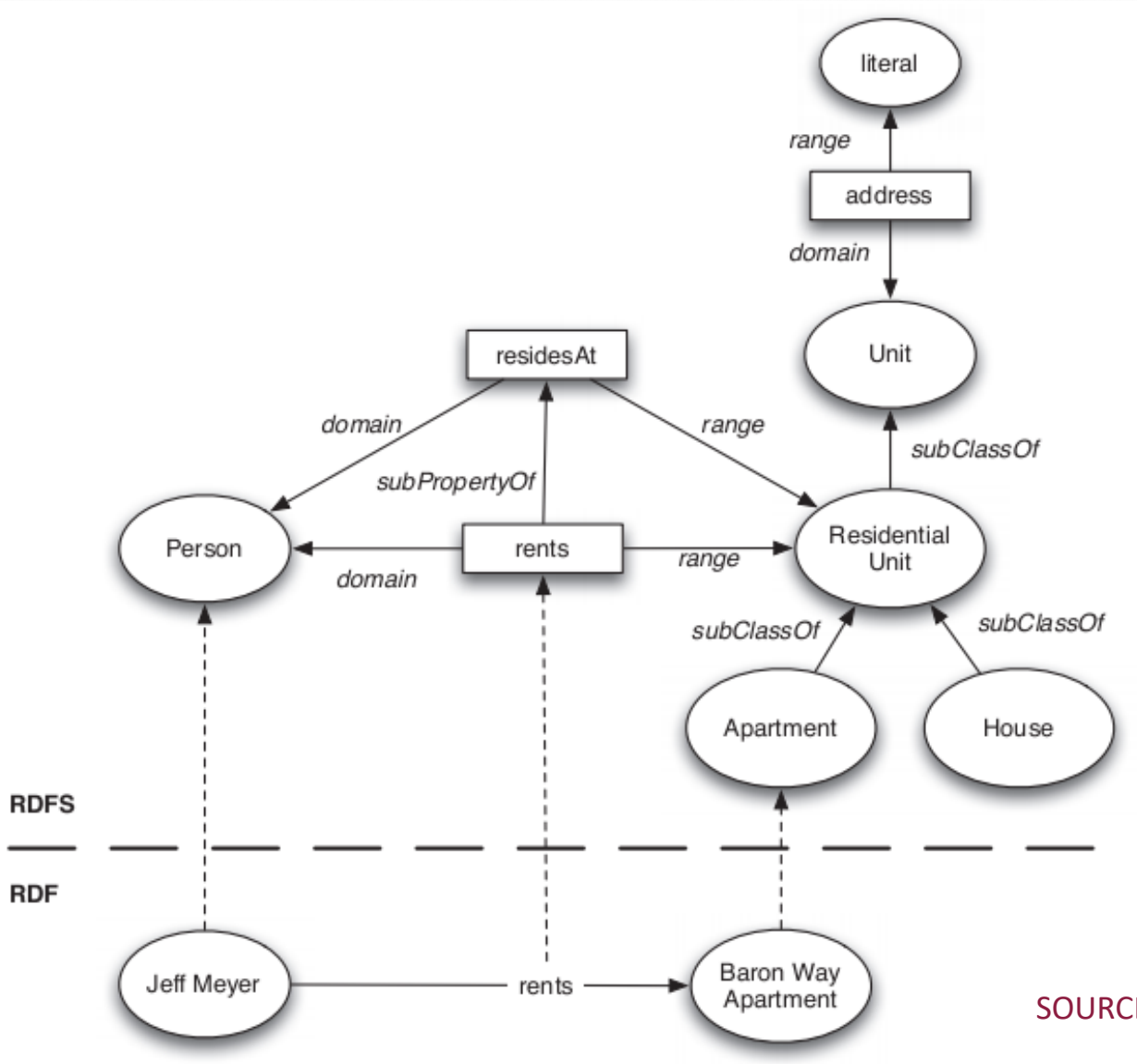
RDF is a universal language that lets users describe resources in their own vocabularies

- RDF does not assume, nor does it define semantics of any particular application domain

The user can do so in RDF Schema using:

- Classes and Properties
- Class Hierarchies and Inheritance
- Property Hierarchies

RDF Layer vs. RDF Schema Layer



SOURCE: “A Semantic Web Primer” by Antoniou, Groth, van Harmelen, Hoekstra

Classes and Instances

- **We must distinguish between**
  - Concrete “things” (individual objects) in the domain:CSE579, Joohyung, etc.
  - Sets of individuals sharing properties called classes: instructors, students, courses, etc.
- **Individual objects that belong to a class are referred to as instances of that class**
- **The relationship between instances and classes in RDF is through rdf:type**

Nonsensical Statements Disallowed by the Use of Classes

CSE579 is taught by CSE259

- We want courses to be taught by academic staff only
- Restriction on values of the property “is taught by” (**range restriction**)

Room 210 is taught by Joohyung

- Only courses can be taught
- This imposes a restriction on the objects to which the property can be applied (**domain restriction**)

Class Hierachies

Classes can be organized in hierarchies

- A is a subclass of B if every instance of A is also an instance of B
- Then B is a superclass of A

A subclass graph need not be a tree

A class may have multiple superclasses

Instances in Class Hiearchies

Range restriction:

- Courses must be taught by academic staff members only
- Joohyung is a professor
- He inherits the ability to teach from the class of academic staff members

This is done in RDF Schema by fixing the semantics of “is a subclass of”

- It is not up to an application (RDF processing software) to interpret “is a subclass of”

Property Hiearchies

Hierarchical relationships for properties

- E.g., “is taught by” is a subproperty of “involves”
- If a course C is taught by an academic staff member A, then C also involves A

The converse is not necessarily true

- E.g., A may be the teacher of the course C, or
- a tutor who marks student homework but does not teach C

P is a subproperty of Q, if Q(x,y) is true whenever P(x,y) is true

Core Classes (1 of 3)

- rdfs:Resource, the class of all resources
- rdfs:Class, the class of all classes
- rdfs:Literal, the class of all literals (strings)
- rdf:Property, the class of all properties.
- rdf:Statement, the class of all reified statements

Core Properties (2 of 3)

- **rdf:type, which relates a resource to its class**
  - The resource is declared to be an instance of that class
- **rdfs:subClassOf, which relates a class to one of its superclasses**
  - All instances of a class are instances of its superclass
- **rdfs:subPropertyOf, relates a property to one of its superproperties**

Core Properties (3 of 3)

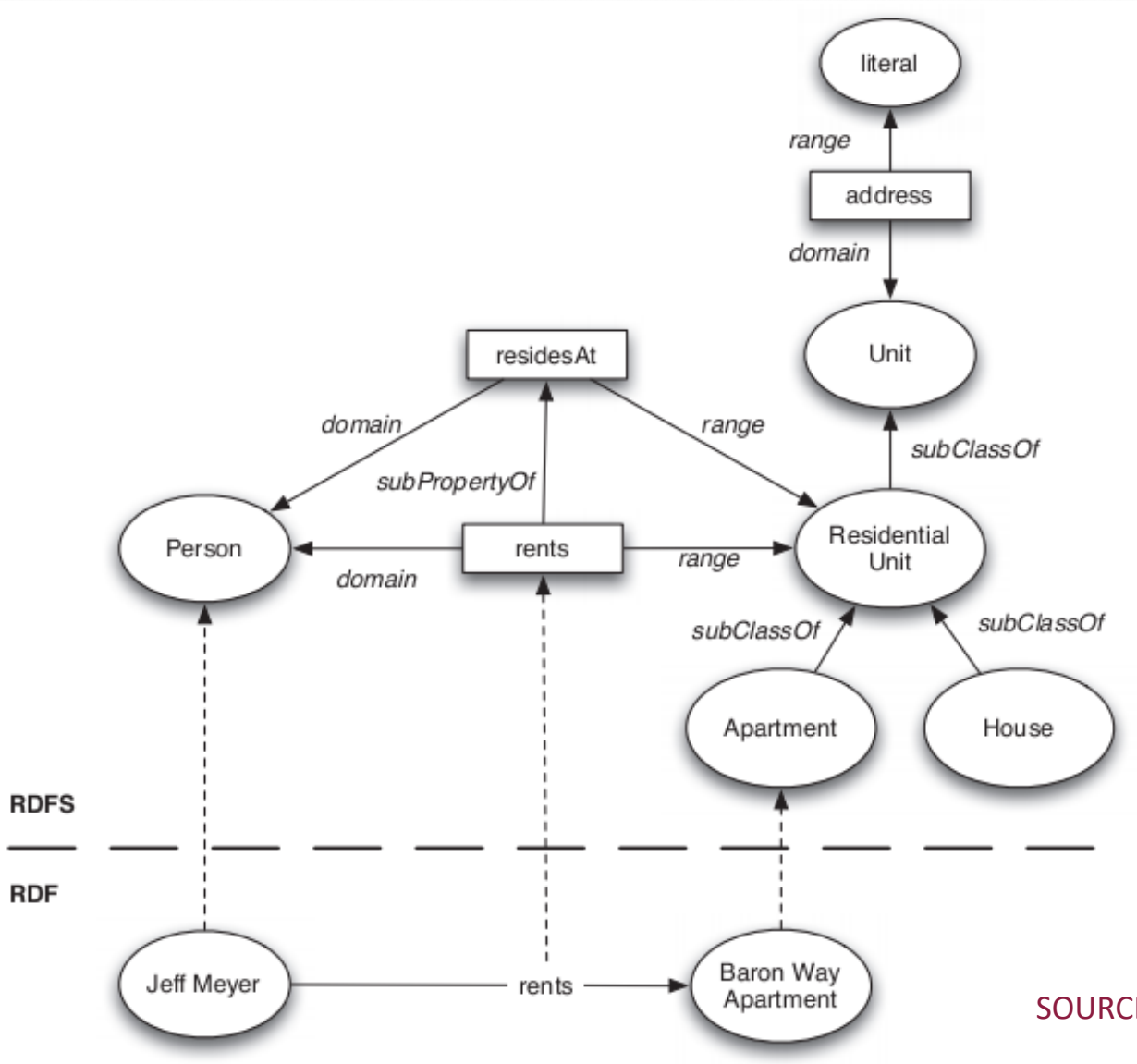
rdfs:domain, which specifies the domain of a property P

- The class of those resources that may appear as subjects in a triple with predicate P
- If the domain is not specified, then any resource can be the subject

rdfs:range, which specifies the range of a property P

- The class of those resources that may appear as values in a triple with predicate P

RDF Layer vs. RDF Schema Layer



SOURCE: “A Semantic Web Primer” by Antoniou, Groth, van Harmelen, Hoekstra

Limitation of RDFS

- It does not support negation; e.g. we could say data:ESI is an org:SME but cannot express that data:ESI is not a person in RDF.

- It does not provide constructors to define classes; e.g. we cannot define what org:SME is in RDFS.

- Although it provides schema-level alignments (with rdfs:subClassOf and rdfs:subPropertyOf), it does not support instance-level alignment; e.g. it cannot express that data:ESI is the same as db:ESI.

Axiomatic Semantics of RDF and RDFS

(From “A Semantic Web Primer” by Antoniou, Groth, van Harmelen, Hoekstra)

Axiomatic Semantics

We formalize the meaning of the modeling primitives of RDF and RDF Schema by translating into first-order logic

- Classes: rdfs:Resource, rdfs:Class, rdf:Property
- Properties: rdf:type, rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, rdfs:range

We make the semantics unambiguous and machine accessible

We provide a basis for reasoning support by automated reasoners manipulating logical formulas

The Approach

- **All language primitives in RDF and RDF Schema are represented by constants:**
  - rdfs:Resource, rdfs:Class, rdf:Property
  - rdf:type, rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, rdfs:range
- **A few predefined predicates are used as a foundation for expressing relationships between the constants**

Basic Predicates

Prop(R,P,V)

- A predicate with 3 arguments, which is used to represent an RDF statement with resource **R**, property **P** and value **V**
- An RDF statement (triple) **(R,P,V)** is represented as **Prop(R,P,V)**.

Type(R,T)

- Short for **Prop(R, rdf:type, T)**
- Specifies that the resource **R** has the type **T**

Type(r, t) $\leftrightarrow$ Prop(r, rdf:type, t)

RDF Classes

- Resource is the most general class: every class and every property is a resource

  Type(p, rdf:Property) $\rightarrow$ Type(p, rdfs:Resource)

  Type(c, rdfs:Class) $\rightarrow$ Type(c, rdfs:Resource)

- The predicate in an RDF statement must be a property

  Prop(r,p,v) $\rightarrow$ Type(p,rdfs:Property)

The rdf:type Property

- rdf:type is a property

  Prop(rdf:type, rdf:type, rdf:Property)

- rdf:type can be applied to resources (domain) and has a class as its value (range)

  Type(r,c) $\rightarrow$ (Type(r, rdfs:Resource) $\wedge$ Type(c, rdfs:Class))

The rdfs:subclassOf property

rdfs:subClassOf is a property:

Type(rdfs:subClassOf, rdf:Property)

If a class C is a subclass of a class D, then all instances of C are also instances of D:

$$\text{Prop}(c, \text{rdfs:subClassOf}, d) \leftrightarrow$$
$$(\text{Type}(c, \text{rdfs:Class}) \wedge \text{Type}(d, \text{rdfs:Class}) \wedge$$
$$\forall x\ (\text{Type}(x, c) \rightarrow \text{Type}(x, d)))$$

The rdfs:subPropertyOf property

P is a subproperty of Q, if Q(x,y) is true whenever P(x,y) is true:

Type(rdfs:subPropertyOf, rdf:Property)

Prop(p, rdfs:subPropertyOf, q) $\leftrightarrow$

(Type(p,rdf:Property) $\wedge$ Type(q,rdf:Property) $\wedge$

$\forall x \; \forall y$ (Prop(x, p, y) $\rightarrow$ Prop(x, q, y)))

The rdfs:domain, rdfs:range Property

- If the domain of P is D, then for every P(x,y), x∈D

$$\text{Prop}(p,\text{rdfs:domain},d) \rightarrow \forall x\ \forall y(\text{Prop}(x,p,y) \rightarrow \text{Type}(x,d))$$

- If the range of P is R, then for every P(x,y), y∈R

$$\text{Prop}(p,\text{ rdfs:range},\ r) \rightarrow \forall x\ \forall y\ (\text{Prop}(x,p,y) \rightarrow \text{Type}(y,r))$$

Wrap-Up