Semantic Web Technologies

Pierre Senellart

23 September 2013, *Università degli Studi Roma Tre*

Outline

Outline

Outline

The Semantic Web

- A Web in which the resources are semantically described
  - annotations give information about a page, explain an expression in a page, etc.
- More precisely, a resource is anything that can be referred to by a URI
  - a web page, identified by a URL
  - a fragment of an XML document, identified by an element node of the document,
  - a web service,
  - a thing, an object, a concept, a property, etc.
- Semantic annotations: logical assertions that relate resources to some terms in associated ontologies

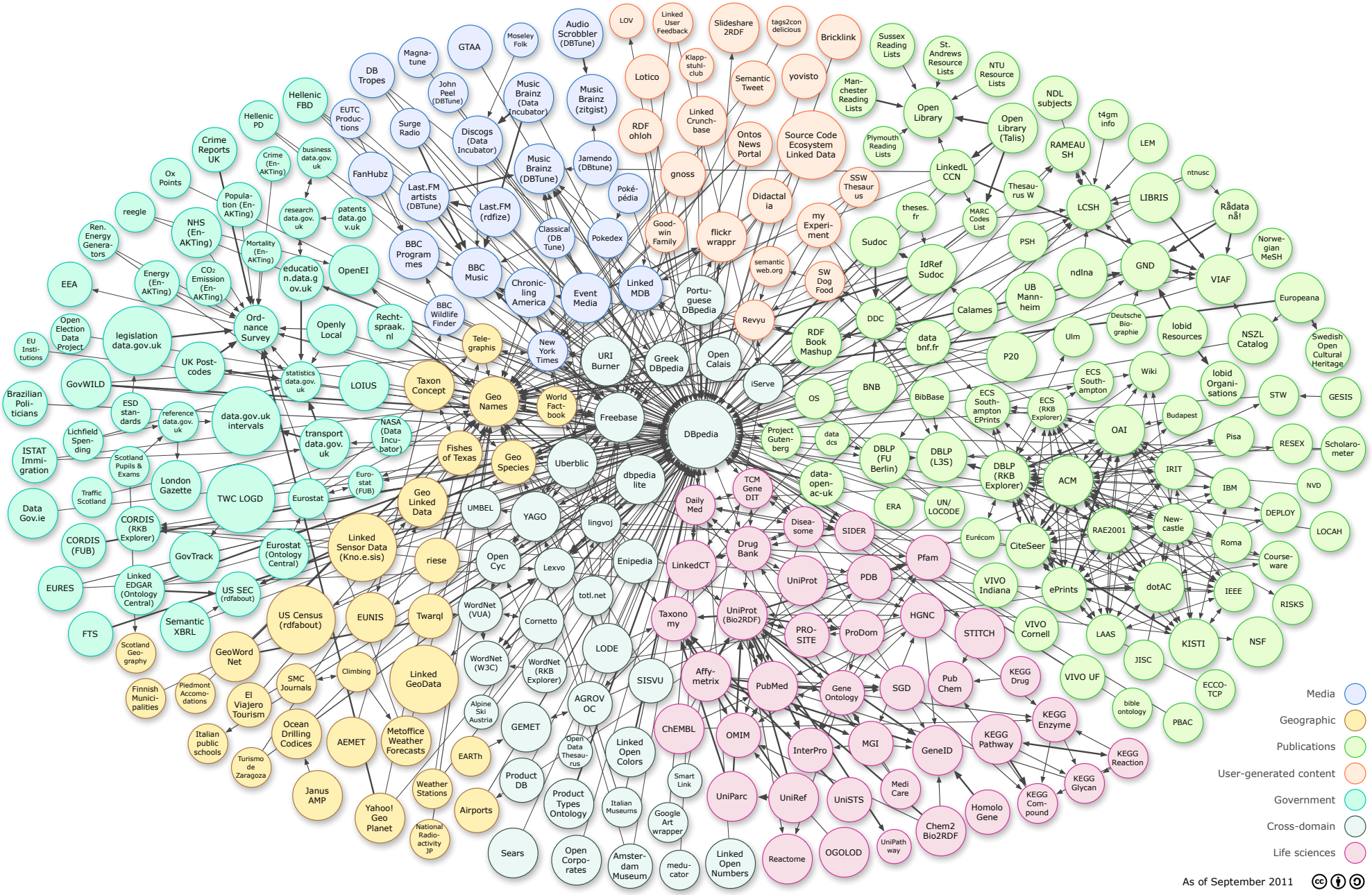
Outline

Ontologies

- Formal descriptions providing human users a shared understanding of a given domain
  - A controlled vocabulary
- Formally defined so that it can also be processed by machines
- Logical semantics that enables reasoning
- Reasoning is the key for different important tasks of Web data management, in particular:
  - to answer queries (over possibly distributed data)
  - to relate objects in different data sources enabling their integration
  - to detect inconsistencies or redundancies
  - to refine queries with too many answers, or to relax queries with no answer

Where Do Ontologies Come From?

- Manually crafted to represent the knowledge of a specific domain (e.g., life sciences)
- Exported from classical Web databases
- Through information extraction from the Web, Wikipedia, etc. (e.g., DBpedia, YAGO)
- Private to a company or public
- Some ontologies focus on instances, others on a schema (see further)
- Value of the Semantic Web: bits of ontologies can be re-used in another, and ontologies can be mapped through an `owl:sameAs` link

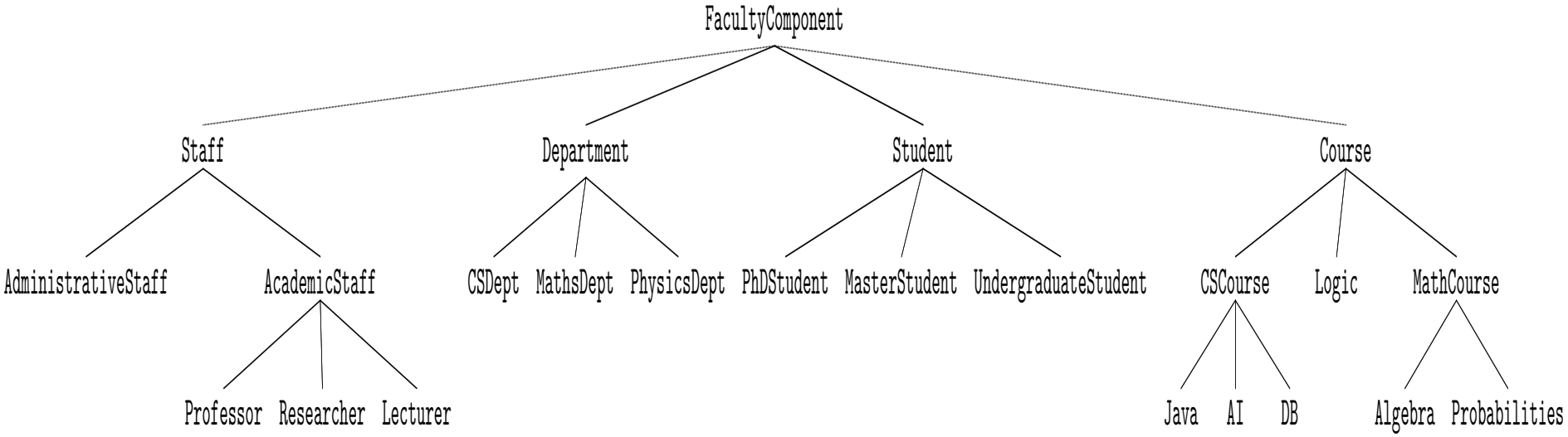


Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

Outline

Classes and class hierarchy

- Backbone of the ontology
- AcademicStaff is a Class (A class will be interpreted as a set of objects)
- AcademicStaff isa Staff (isa is interpreted as set inclusion)



Relations

- Declaration of relations with their signature
- (Relations will be interpreted as binary relations between objects)
- `TeachesIn(AcademicStaff, Course)`
  - if one states that "X `TeachesIn` Y", then X belongs to `AcademicStaff` and Y to `Course`
- `TeachesTo(AcademicStaff, Student)`
- `Leads(Staff, Department)`

Instances

- Classes have instances
- Dupond is an instance of the class `Professor`
- corresponds to the fact: `Professor(Dupond)`

- Relations also have instances
- (Dupond,CS101) is an instance of the relation `TeachesIn`
- corresponds to the fact: `TeachesIn(Dupond,CS101)`

- The instance statements can be seen as (and stored in) a database

Ontology = schema + instance

- Schema (TBox)
  - The set of class and relation names
  - The signatures of relations and also constraints
  - The constraints are used for two purposes
    - checking data consistency (like dependencies in databases)
    - inferring new facts
- Instance (ABox)
  - The set of facts
  - The set of base facts together with the inferred facts should satisfy the constraints
- Ontology (i.e., Knowledge Base) = Schema + Instance

Outline

3 ontology languages for the Web

- RDF: a very simple ontology language
- RDFS: Schema for RDF
  - Can be used to define richer ontologies
- OWL: a much richer ontology language

- We next present them rapidly

Outline

Namespaces

- A URI most often takes the form of a URL:
  - `http://live.dbpedia.org/page/%C3%89lectricit%C3%A9_de_France`
  - `http://sw.opencyc.org/2012/05/10/concept/en/GameOfChance`
  - `http://www.w3.org/People/Berners-Lee/card#i`
  - `http://www4.wiwiss.fu-berlin.de/dblp/resource/record/conf/www/2006`
- Some of these URIs may be actual URLs (point to a browsable resource), some may just be abstract
- For simplicity, possibility of defining namespace prefixes, e.g., `dbpedia = http://live.dbpedia.org/page/` along with a default prefix (e.g., mapped to `http://sw.opencyc.org/2012/05/10/concept/en/`):
  - `dbpedia:%C3%89lectricit%C3%A9_de_France`
  - `:GameOfChance`

RDF: Resource Description Framework

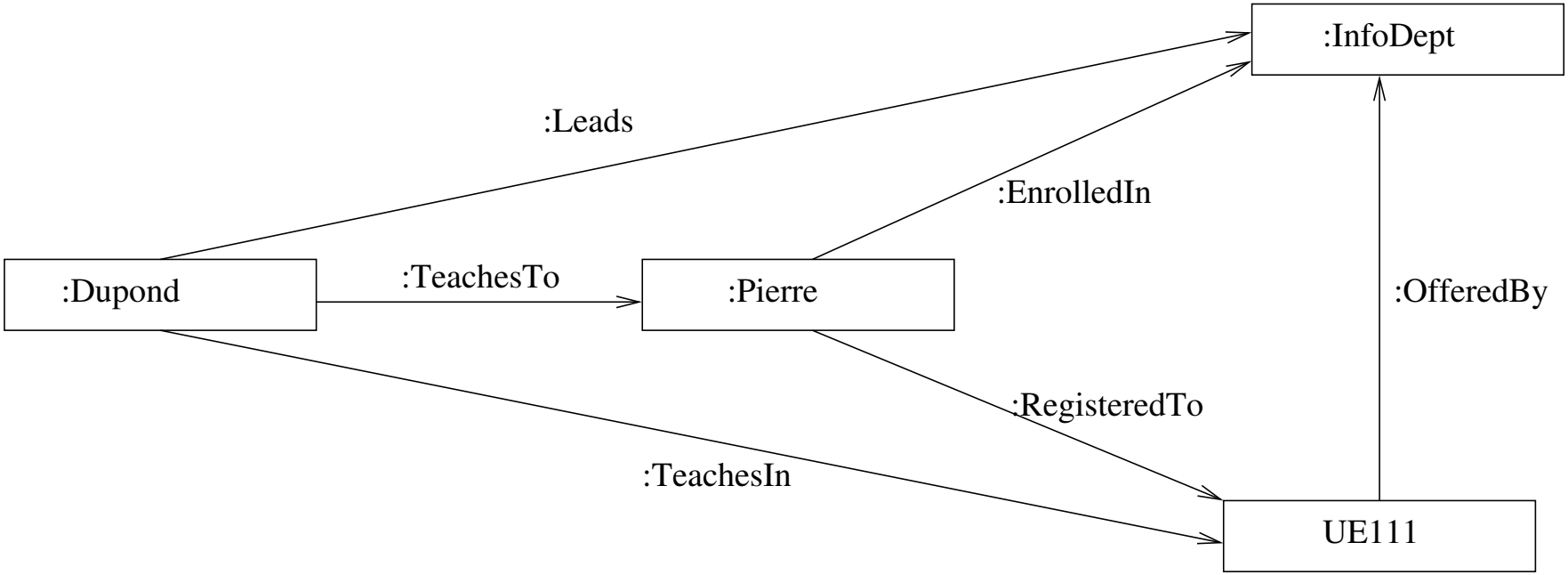
- RDF facts are triplets
- Each triplet is of the form: ⟨ Subject Predicate Object ⟩
- The subject is a URI, referencing an entity
- The predicate is a URI, referencing a relation
- The object is either a URI, referencing an entity, or a literal

```
⟨ :Dupond :Leads :CSDept ⟩
⟨ :Dupond :HasName "Paul Dupond" ⟩
⟨ :Dupond :TeachesIn :UE111 ⟩
⟨ :Dupond :TeachesTo :Pierre ⟩
⟨ :Pierre :EnrolledIn :CSDept ⟩
⟨ :Pierre :RegisteredTo :UE111 ⟩
⟨ :UE111 :OfferedBy :CSDept ⟩
```

- The linked data cloud contains dozens of billions of RDF triples

RDF graph

- A set of RDF facts defines
  - a set of relations between objects
  - an RDF graph where the nodes are objects:



RDF semantics

- A triplet $\langle s \;\; P \;\; o \rangle$ is interpreted in first-order logic (FOL) as a fact $P(s, o)$
- Example:

  `Leads(Dupond, CSDept)`
  `TeachesIn(Dupond, UE111)`
  `TeachesTo(Dupond,Pierre)`
  `EnrolledIn(Pierre, CSDept)`
  `RegisteredTo(Pierre, UE111)`
  `OfferedBy(UE111, CSDept)`

RDF Serialization

Several serialization formats for RDF data, see alternate formats of `http://live.dbpedia.org/page/%C3%89lectricit%C3%A9_de_France`:

- RDF/XML, structured XML representation, allowing for nesting
- N3 or Turtle, more compact, readable syntax, with some syntaxic sugar (Turtle is a subset of N3)
- N-Triples, a simple text-based format
- RDFa to integrate RDF annotations into HTML

Outline

RDFS: RDF Schema

- Not detailed here: the schema in RDF is super simplistic
- An RDF Schema defines the schema of a richer ontology

RDF Schema

- Do net get confused: RDFS can use RDF as syntax, i.e., RDFS statements can be themselves expressed as RDF triplets using some specific properties and objects used as RDFS keywords with a particular meaning.

- Declaration of classes and subclass relationships
  - ⟨ `Staff` `rdf:type` `rdfs:Class` ⟩
  - ⟨ `Java` `rdfs:subClassOf` `CSCourse` ⟩
- Declaration of instances (beyond the pure schema)
  - ⟨ `Dupond` `rdf:type` `AcademicStaff` ⟩

RDF Schema - continued

- Declaration of relations (properties in RDFS terminology)
  - ⟨ `RegisteredTo` `rdf:type` `rdfs:Property` ⟩
- Declaration of subproperty relationships
  - ⟨ `LateRegisteredTo` `rdfs:subPropertyOf` `RegisteredTo` ⟩
- Declaration of domain and range restrictions for predicates
  - ⟨ `TeachesIn` `rdfs:domain` `AcademicStaff` ⟩
  - ⟨ `TeachesIn` `rdfs:range` `Course` ⟩
  - `TeachesIn`(`AcademicStaff`, `Course`)

RDFS logical semantics

RDF and RDFS statements	FOL translation	DL notation
⟨ i rdf:type C ⟩	$C(i)$	$i : C$ or $C(i)$
⟨ i P j ⟩	$P(i, j)$	$i\, P\, j$ or $P(i, j)$
⟨ C rdfs:subClassOf D ⟩	$\forall X\, (C(X) \Rightarrow D(X))$	$C \sqsubseteq D$
⟨ P rdfs:subPropertyOf R ⟩	$\forall X\, \forall Y\, (P(X, Y) \Rightarrow R(X, Y))$	$P \sqsubseteq R$
⟨ P rdfs:domain C ⟩	$\forall X\, \forall Y\, (P(X, Y) \Rightarrow C(X))$	$\exists P \sqsubseteq C$
⟨ P rdfs:range D ⟩	$\forall X\, \forall Y\, (P(X, Y) \Rightarrow D(Y))$	$\exists P^{-} \sqsubseteq D$

DL: Description logics, a specialized logical formalism

Outline

OWL: Web Ontology Language

- OWL extends RDFS with the possibility to express additional constraints
- Main OWL constructs
  - Disjointness between classes
  - Constraints of functionality and symmetry on predicates
  - Intentional class definitions
  - Class union and intersection
- Inspired by description logics
- Several profiles: OWL Full, OWL DL, OWL Lite, OWL 2 EL, OWL 2 QL, OWL 2 RL. Different profiles include different features, and have different tractability

OWL constructs

- Disjointness between classes:

OWL notation	FOL translation	DL notation
⟨ C `owl:disjointWith` D ⟩	$\forall X\,(C(X) \Rightarrow \neg D(X))$	$C \sqsubseteq \neg D$

- Constraints of functionality and symmetry on predicates:

OWL notation	FOL translation	DL notation
⟨ P `rdf:type` `owl:FunctionalProperty` ⟩	$\forall X\,\forall Y\,\forall Z$ $(P(X,Y) \wedge P(X,Z) \Rightarrow Y = Z)$	$(\mathit{funct}\,P)$ or $\exists P \sqsubseteq (\leq 1\,P)$
⟨ P `rdf:type` `owl:InverseFunctionalProperty` ⟩	$\forall X\,\forall Y\,\forall Z$ $(P(X,Y) \wedge P(Z,Y) \Rightarrow X = Z)$	$(\mathit{funct}\,P^-)$ or $\exists P^- \sqsubseteq (\leq 1\,P^-)$
⟨ P `owl:inverseOf` Q ⟩	$\forall X\,\forall Y\,(P(X,Y) \Leftrightarrow Q(Y,X))$	$P \equiv Q^-$
⟨ P `rdf:type` `owl:SymmetricProperty` ⟩	$\forall X\,\forall Y\,(P(X,Y) \Rightarrow P(Y,X))$	$P \sqsubseteq P^-$

Definition of intentional classes in OWL

- Goal: allow expressing complex constraints such as:
  - departments can be lead only by professors
  - only professors or lecturers may teach to undergraduate students.
- The keyword `owl:Restriction` is used in association with a blank node class, and some specific restriction properties:
  - `owl:someValuesFrom`
  - `owl:allValuesFrom`
  - `owl:minCardinality`
  - `owl:maxCardinality`

OWL Semantics

OWL notation	FOL translation	DL notation
`_a owl:onProperty P` `_a owl:allValuesFrom C`	$\forall Y\,(P(X,Y) \Rightarrow C(Y))$	$\forall P.C$
`_a owl:onProperty P` `_a owl:someValuesFrom C`	$\exists Y\,(P(X,Y) \wedge C(Y))$	$\exists P.C$
`_a owl:onProperty P` `_a owl:minCardinality` n	$\exists Y_1 \ldots \exists Y_n (P(X,Y_1) \wedge \ldots \wedge P(X,Y_n) \wedge \bigwedge_{i,j \in [1..n], i \neq j} (Y_i \neq Y_j))$	$(\geq n\, P)$
`_a owl:maxCardinality` n	$\forall Y_1 \ldots \forall Y_n \forall Y_{n+1}$ $(P(X,Y_1) \wedge \ldots \wedge P(X,Y_n) \wedge P(X,Y_{n+1})$ $\Rightarrow \bigvee_{i,j \in [1..n+1], i \neq j} (Y_i = Y_j))$	$(\leq n\, P)$

Unnamed new classes by example

- Departments can be lead only by professors

- Define the set of objects that are lead by professors

```
_a   rdfs:subClassOf     owl:Restriction
_a   owl:onProperty      Leads
_a   owl:allValuesFrom   Professor
```

- Now specify that all departments are lead by professors

```
Department   rdfs:subClassOf   _a
```

Union and Intersection of Classes by example

- Only professors or lecturers may teach to undergraduate students

```
_a  rdfs:subClassOf      owl:Restriction
_a  owl:onProperty       TeachesTo
_a  owl:someValuesFrom   Undergrad
_b  owl:unionOf          (Professor, Lecturer)
_a  rdfs:subClassOf      _b
```

- This corresponds to an inclusion axiom in Description Logic:

$$\exists\ TeachesTo.UndergraduateStudent \sqsubseteq Professor \sqcup Lecturer$$

- `owl:equivalentClass` corresponds to double inclusion:

$$MathStudent \equiv Student \sqcap \exists\ RegisteredTo.MathCourse$$

Outline

Querying using RDFS

- RDFS statements can be used to infer new triples
- Example
  - Base fact $ResponsibleOf(durand, ue111)$
  - Use the statement $\langle ResponsibleOf$ `rdfs:domain` $Professor\rangle$
    i.e., the logical rule: $ResponsibleOf(X, Y) \Rightarrow Professor(X)$
  - With substitution {X/durand, Y/ue111}
  - Infer fact $Professor(durand)$
  - Use the statement $\langle Professor$ `rdfs:subClassOf` $AcademicStaff\rangle$
    i.e., the rule $Professor(X) \Rightarrow AcademicStaff(X)$
  - With substitution {X/durand}
  - Infer fact $AcademicStaff(durand)$
  - etc.

The saturation algorithm

- Keep infering new facts until a fixpoint is reached
- Note: Only polynomially many facts can be added
- PTIME

Querying using DL

- RDFS simpler and very used but limited
- Query answering is unfeasible for all of OWL (even OWL-DL)... but restrictions exist that have more reasonable complexity (OWL-Lite, OWL 2 QL, OWL 2 EL). OWL-Full is even undecidable!

SPARQL: query language for RDF data

- Graph pattern language
- Get all URIs, name, emails of persons with all three pieces of information from a FOAF dataset:

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT *
WHERE {
    ?person foaf:name ?name .
    ?person foaf:mbox ?emails .
}
```

- The SPARQL processor may or may not use reasoning w.r.t. the schema

Outline

Where can Semantic Content be Found?

- In the linked data, through Web-available RDF data:
  - dumps of an entire ontology, in one of the RDF serialization formats (RDF/XML, Turtle, N-Triples)
  - crawlable RDF content, with small fragments pointing to other fragments
  - a SPARQL endpoint
  - HTML annotated with RDFa, cf. `http://www.w3.org/TR/rdfa-syntax/`
- Other popular semantic content embedded in Web pages: microformats (hCard, vCard, etc.), microdata (cf. `http://schema.org/`). Not directly the spirit of the Semantic Web, but heavily used.
- RDF content used internally in a company

Systems

- RDF stores (triplestores) with relational or native backend, open-source or commercial, see http://en.wikipedia.org/wiki/Triplestore. Apache Jena is a popular open-source Java store.
- SPARQL engines, usually on top of a triplestore. http://en.wikipedia.org/wiki/SPARQL
- Semantic browsers, to navigate RDF content, http://en.wikipedia.org/wiki/Linked_data#Browsers
- A semantic Web search engine, http://sig.ma/
- Tool to view semantic data in Web pages: http://www.google.com/webmasters/tools/richsnippets

To go further...

- The specifications on `http://www.w3.org/`
- A gentle introduction to the Semantic Web:
  A Semantic Web Primer, G. Antoniou, F. van Harmelen, MIT Press, 2008
- A practical approach:
  Semantic Web for the Working Ontologist: Modeling in RDF, RDFS and OWL, Dean Allemang, James A. Hendler, Morgan-Kaufman, 2008
- Contents of this lecture from:
  Web data management, S. Abiteboul, I. Manolescu, P. Rigaux, M.-C. Rousset, P. Senellart, Cambridge University Press, 2012.
  Also available at `http://webdam.inria.fr/Jorge/`