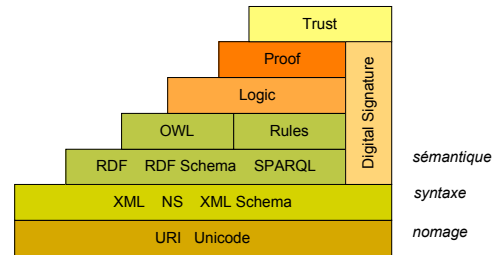


Les Langages du Web Sémantique

Catherine Faron Zucker
faron@polytech.unice.fr

1

The Semantic Web in a Nutshell



INSA Bourges

Catherine Faron Zucker

2

RDF Vocabulary Description Langage: RDF Schema

<http://www.w3.org/TR/rdf-schema/>

3

RDF et RDF Schema

- RDF sert à annoter des ressources
- Ces annotations reposent sur des vocabulaires partagés : des ontologies
- RDF Schema permet de définir *en RDF* des vocabulaires RDF
- RDF Schema est un ensemble de classes et de propriétés RDF qui peuvent être utilisées pour définir des classes et des propriétés (en RDF)

INSA Bourges

Catherine Faron Zucker

4

Une approche centrée sur les propriétés

- RDF diffère des systèmes de typage des langages à objets en ce sens que : plutôt que de définir une classe en terme de propriétés que possèdent ses instances, *RDFS décrit une propriété en terme de classes de ressources auxquelles elle s'applique*

```
public class Book {  
    private String author;  
    ...  
}  
  
ns:author rdf:type rdf:Property  
ns:author rdfs:domain ns:Document  
ns:author rdfs:range ns:Person
```

INSA Bourges

Catherine Faron Zucker

5

Signature d'une propriété (1/2)

- Domain (ou conjonction de domaines) + Range
 - Une ressource peut être instance de plusieurs classes

```
ns:vitesse rdf:type rdf:Property  
ns:vitesse rdfs:domain ns:Objet  
ns:vitesse rdfs:domain ns:Mobile  
ns:vitesse rdfs:range rdfs:Literal
```

INSA Bourges

Catherine Faron Zucker

6

Signature d'une propriété (2/2)

■ Typage

```
swl:enseignant rdf:type rdf:Property
swl:enseignant rdfs:domain swl:Cours
swl:enseignant rdfs:range swl:Person
_:x swl:enseignant swl:catherine
-----
_:x rdf:type swl:Cours
swl:catherine rdf:type swl:Person
```

Hiérarchies de classes et de propriétés

■ ns:Book **rdfs:subClassOf** ns:Document

■ ns:author **rdfs:subPropertyOf** ns:actor

■ Héritage multiple

ns:EnseignantChercheur **rdfs:subClassOf** ns:Enseignant
ns:EnseignantChercheur **rdfs:subClassOf** ns:Chercheur

Hiérarchies de classes et de propriétés

■ Typage

```
rdf:type rdfs:range rdfs:Class
ex:olivier rdf:type ex:Person
⇒ ex:Person rdf:type rdfs:Class

ex:olivier rdf:type ex:Researcher
ex:Researcher rdfs:subClassOf ex:Person
⇒ ex:olivier rdf:type ex:Person

ex:olivier ex:responsableDeCours ex:log3
ex:responsableDeCours rdfs:subPropertyOf ex:enseignant
⇒ ex:olivier ns:enseignant ex:log3
```

Signatures de propriétés et héritage

Man subClassOf Primate
Chimpanzee subClassOf Primate

```
<rdf:Property rdf:ID='child'>
  <rdfs:domain rdf:resource='#Primate'/>
  <rdfs:range rdf:resource='#Primate'/>
</rdf:Property>
```

```
<rdf:Property rdf:ID='manChild'>
  <rdfs:subPropertyOf rdf:resource='#child'/>
  <rdfs:domain rdf:resource='#Man'/>
  <rdfs:range rdf:resource='#Man'/>
</rdf:Property>
```

Syntaxe RDF/XML (1/2)

```
<rdf:Property rdf:ID='vitesse'>
  <rdfs:domain rdf:resource='#Objet'/>
  <rdfs:domain rdf:resource='#Mobile'/>
  <rdfs:range rdf:resource='&rdfs:Literal'/>
</rdf:Property>

<rdf:Description rdf:ID='vitesse'>
  <rdf:type>
    <rdf:Description rdf:about='http://...Property'/>
  </rdf:type>
  <rdf:domain>
    <rdf:Description rdf:about='#Objet'/>
  </rdf:domain>
  ...
</rdf:Description>
```

Syntaxe RDF/XML (2/2)

```
<rdf:Class rdf:ID='Researcher'>
  <rdfs:subClassOf rdf:resource='#Person'/>
</rdf:Class>

<rdf:Property rdf:ID='responsableDeCours'>
  <rdfs:subPropertyOf rdf:resource='#enseignant'/>
</rdf:Property>
```

Une annotation basée sur une ontologie

```
<rdf:Description rdf:about='olivier'>
  <rdf:type rdf:resource='#Chercheur'/>
  <ns:institut>INRIA</ns:institut>
  <swl:responsableDeCours>
    <Cours rdf:resource='http://courswiki.essi.fr...'>
  </swl:responsableDeCours>
  <rdf:type rdf:resource='#Enseignant'/>
  <ns:etablissement>EPU</ns:etablissement>
</rdf:Description>
```

Cohabitation de plusieurs schémas

- Grâce aux namespaces XML

```
<rdf:Description
  xmlns:rdf='&rdf;'
  xmlns:ns='http://www.inria.fr/acacia/cours#'
  xmlns:swl='http://courswiki.essi.fr/LOG/SemanticWebLangages#'>

  <ns:institut>INRIA</ns:institut>
  <swl:etablissement>EPU</swl:etablissement>
</rdf:Description>
```

Rdfs:label

- Une ressource peut avoir un ou plusieurs noms externes appelés labels
- Les labels sont multilingues

```
<rdf:Property rdf:ID='name'>
  <rdfs:domain rdf:resource='Person'/>
  <rdfs:range rdf:resource='&rdfs:Literal'/>

  <rdfs:label xml:lang='fr'>nom</rdfs:label>
  <rdfs:label xml:lang='en'>name</rdfs:label>
</rdf:Property>
```

Rdfs:comment

- Définition en langage naturel

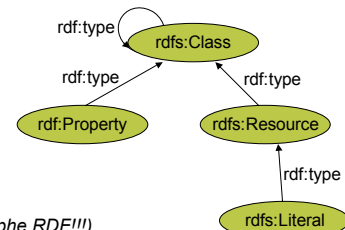
```
<rdfs:Class rdf:about='#Thing'>
  <rdfs:comment xml:lang='fr'>La racine de la hiérarchie des
  classes</rdfs:comment>
  <rdfs:comment xml:lang='en'>The root of the class
  hierarchy</rdfs:comment>
</rdfs:Class>
```

Rdfs:seeAlso

```
<rdfs:Class rdf:about='#Man'>
  <rdfs:seeAlso rdf:resource='#Person'/>
</rdfs:Class>
```

Méta-modèle RDF

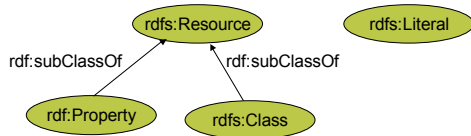
- **rdfs:Class**: méta-classe qui engendre les classes, instance d'elle-même



(ceci est un graphe RDF!!!)

Méta-modèle RDF

- **rdfs:Resource**: classe racine de la hiérarchie de subsomption des classes

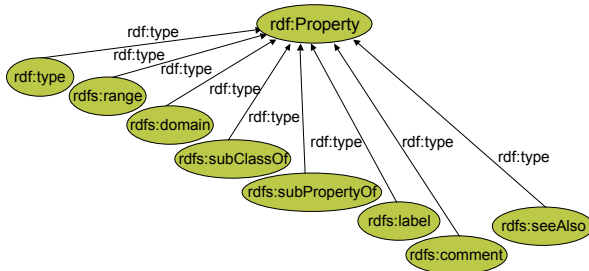


(ceci est un graphe RDF!!!)

Méta-modèle RDF

- Modèle standard de bootstrap ObjVLisp
- **rdfs:Resource**: classe racine de la hiérarchie de subsomption des classes, instance de rdfs:Class
- **rdfs:Class**: méta-classe qui engendre les classes, instance d'elle-même, sous-classe de rdfs:Resource
- rdfs:Resource rdf:type rdfs:Class
- rdfs:Class rdf:type rdfs:Class
- rdfs:Class rdfs:subClassOf rdfs:Resource

Méta-modèle RDF



(ceci est un graphe RDF!!!)

Méta-modèle RDF

- **rdf:Property** La classe des propriétés
- rdf:Property rdf:type rdfs:Class
- rdf:Property rdfs:subClassOf rdfs:Resource

Méta-modèle RDF

- **rdf:type** relation d'instanciation
rdf:type rdf:type rdf:Property
- **rdfs:subClassOf** relation de subsomption
rdfs:subClassOf rdf:type rdf:Property
- **rdfs:subPropertyOf** relation de subsomption
rdfs:subPropertyOf rdf:type rdf:Property

Extension du méta-modèle RDF (1/2)

- ex:isa** rdfs:subPropertyOf rdf:type
ex:ma4L ex:isa ex:Car
- ex:extends** rdfs:subPropertyOf rdfs:subClassOf
ex:Car ex:extends ex:Mobile
- ex:Concept** ex:extends rdfs:Class
ex:Car ex:isa ex:Concept

Extension du méta-modèle RDF (2/2)

ex:Relation rdfs:subClassOf rdf:Property
ex:hasDomain rdfs:subPropertyOf rdfs:domain
ex:valueType rdfs:subPropertyOf rdfs:range

ex:speed ex:isa ex:Relation
ex:speed ex:hasDomain ex:Mobile
ex:speed ex:valueType xsd:float

Exemple de Schema RDFS (1/2)

```
<rdfs:Class rdf:ID="Document">
<rdfs:subClassOf rdf:resource="#Entity"/>
<rdfs:subClassOf rdf:resource="#EntityConcerningATopic"/>
<rdfs:subClassOf rdf:resource="#NumberableEntity"/>
<rdfs:comment xml:lang="en">Entity including elements serving as
a representation of thinking.</rdfs:comment>
<rdfs:comment xml:lang="fr">Entite comprenant des elements de
representation de la pensee.</rdfs:comment>
<rdfs:label xml:lang="en">document</rdfs:label>
<rdfs:label xml:lang="fr">document</rdfs:label>
</rdfs:Class>
```

Exemple de Schema RDFS (2/2)

```
<rdf:Property rdf:ID="Designation">
<rdfs:subPropertyOf rdf:resource="#SomeRelation"/>
<rdfs:range rdf:resource="#&rdfs;Literal"/>
<rdfs:domain rdf:resource="#Something"/>
<rdfs:comment xml:lang="en">Identifying word or words by which a
thing is called and classified or distinguished from
others.</rdfs:comment>
<rdfs:comment xml:lang="fr">Identifiant le ou les mots par lesquels
une chose s appelle et est classifiee ou distinguee des
autres.</rdfs:comment>
<rdfs:label xml:lang="en">designation</rdfs:label>
<rdfs:label xml:lang="fr">designation</rdfs:label>
</rdf:Property>
```

Exemple d'annotation RDF

```
<c:ResearchReport rdf:about="http://www.inria.fr/rapports/sophia/RR-3071.html">
<c:title>Modèles Génériques de Gestion de Conflits dans la
Conception Concourante</c:title>
<c:CreationDate>12-1996</c:CreationDate>
<c:CreatedBy>
<c:Person rdf:about="http://www.inria.fr/acacia/Olivier.Corby">
<c:FirstName>Olivier</c:FirstName>
<c:FamilyName>Corby</c:FamilyName></c:Person>
</c:CreatedBy>
...
<c:CreatedBy>
<c:ProjectGroup
rdf:about="http://www.inria.fr/recherche/equipes/acacia.en.html">
<c:Designation>Acacia</c:Designation>
</c:ProjectGroup>
</c:CreatedBy>
</c:ResearchReport>
```

Autres exemples RDF

- Répertoire de schema : <http://www.schemaweb.info/>
- W3C Composite Capability/Preference Profiles : <http://www.w3.org/TR/CCPP-struct-vocab>
- Site Syndication : <http://web.resource.org/rss/1.0/>
- Dublin Core
- Adobe XMP extensible metadata platform
- Topic Map en RDF
- PRISM: Publishing Requirements for Industry Standard Metadata
- Gene Ontology
- RDF binding IEEE LOM Learning Object Metadata
- Ontology Development Metamodel (binding UML RDF OWL) www.omg.org/docs/ontology/04-08-01.pdf

Retour sur SPARQL

- RDF data
_x rdf:type swl:Homme
_y rdf:type swl:Femme
- RDFS data
swl:Homme rdfs:subClassOf swl:HumanBeing
swl:Femme rdfs:subClassOf swl:HumanBeing
- SPARQL
SELECT _h WHERE { _h rdf:type swl:HumanBeing }

Retour sur SPARQL

■ RDF(S) data

```
swl:Homme rdfs:subClassOf swl:HumanBeing
swl:Femme rdfs:subClassOf swl:HumanBeing
swl:HumanBeing rdfs:type rdfs:Class
```

■ SPARQL

```
SELECT _c WHERE { _c rdfs:type swl:Class }
```

Retour sur la projection de graphes

- G subsume G' ssi il existe une projection de G dans G'
- Une projection de G dans G' est une application π des sommets de G dans ceux de G' telle que:
 - Pour tout sommet x de G, $e(x) \geq \pi(e(x))$
 - Pour tout arc $\langle s, p, o \rangle$ de G, $\langle \pi(s), \pi(p), \pi(o) \rangle$ est arc de G'

Recherche approchée

■ Offerte par CORESE

□ Distance ontologique

Une projection approchée de G dans G' est une application π des sommets de G dans ceux de G' telle que:

- Pour tout sommet x de G,
 $e(x)$ est proche de $\pi(e(x))$ dans l'ontologie
- Pour tout arc $\langle s, p, o \rangle$ de G,
 $\langle \pi(s), \pi(p), \pi(o) \rangle$ est arc de G'

□ Chemin de relations

Règles ontologiques

If ... Then ...

- Compléter une base RDF à l'aide de règles définies dans l'ontologie
- Application des règles sur la base RDF
 - en chaînage avant
 - à saturation
- Condition d'application
 - Il existe une projection du graphe en prémisse de la règle sur la base RDF

Règles usuelles

- Transitivité
 - subClassOf, subPropertyOf, ancetreDe
- Symétrie
 - marriedWith
- Propriétés inverses
 - parentDe inverse de enfantDe

Règles liées au domaine

- Si un membre d'une équipe de recherche a un centre d'intérêt, alors cette équipe a aussi ce centre d'intérêt

```
?person interestedBy ?topic
?person member ?researchTeam
-----
?team interestedBy ?topic
```

Web Ontology Language OWL

<http://www.w3.org/2004/OWL/>

OWL in a Nutshell

- OWL « au-dessus » de RDFS
 - `xmlns:owl=http://www.w3.org/2002/07/owl#`
- 3 niveaux d'expressivité
 - OWL Lite
 - OWL DL
 - OWL Full

OWL in a Nutshell

- Propriétés algébriques des propriétés
 - `ex:isMarriedWith` est symétrique
- Alignement d'ontologies
 - `ex1:Voiture` est équivalent à `ex2:Car`
- Contraintes de cohérence
 - `ex:Homme` et `ex:Femme` sont disjointes
- Définition formelle de classes
 - `ex:Manager(?x) ssi ?x manage ?y`
- Restrictions des propriétés et raffinement
 - Le range de `ex:child` devient `ex:Human` pour `ex:Human`

OWL héritier des DL

- Class axioms
- Property axioms
- Facts about individuals
- Logiques de Description (DL)
 - niveau assertionnel (Abox) qui manipule les individus
 - niveau terminologique (Tbox) qui manipule concepts (classes) et rôles (propriétés)

OWL héritier des DL

- Définitions partielles
 - Conditions nécessaires: `rdfs:subClassOf`, `rdfs:subPropertyOf`
`Homme < Personne`
`Femme < (and Personne (not Homme))`
`membre < toprole`
- Définitions complètes
 - Conditions nécessaires et suffisantes: `owl:equivalentClass`
`Equipe = (and Ensemble`
 `(all membre Personne)`
 `(atleast 2 membre))`

OWL héritier des DL

- Classification de concepts et de rôles
- Identification d'instances
- Complexité algorithmique fonction de l'expressivité de la DL considérée
 - Famille de DL
 - D'où les 3 couches de OWL

Descriptions de classe

- Class identifier (URI reference)
- Exhaustive enumeration of individuals that together form the instances of a class
- Property restriction
- Intersection of two or more class descriptions
- Union of two or more class descriptions
- Complement of a class description

Descriptions de classe

- La classe des classes OWL
 - Toutes les descriptions de classe sont typées `owl:Class`
 - `owl:Class` `rdfs:subClassOf` `rdfs:Class`
- Classe nommée
 - `<owl:Class rdf:ID="Human" />`

Descriptions de classe

- Enumeration d'individus: definition en extension

```
<owl:Class>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Eurasia"/>
    <owl:Thing rdf:about="#Africa"/>
    <owl:Thing rdf:about="#NorthAmerica"/>
    <owl:Thing rdf:about="#SouthAmerica"/>
    <owl:Thing rdf:about="#Australia"/>
    <owl:Thing rdf:about="#Antarctica"/>
  </owl:oneOf>
</owl:Class>
```

Descriptions de classe

- Property restriction
 - Value constraints
 - Restriction de toutes les valeurs d'une propriété
- ```
<owl:Restriction>
 <owl:onProperty rdf:resource="#hasParent"/>
 <owl:allValuesFrom rdf:resource="#Human"/>
</owl:Restriction>
```

## Descriptions de classe

- Property restriction
    - Value constraints
      - Restriction d'au moins une valeur d'une propriété
  - Donnée de la valeur d'une propriété
- ```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent"/>
  <owl:someValuesFrom rdf:resource="#Physician"/>
</owl:Restriction>

<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent"/>
  <owl:hasValue rdf:resource="#Sarkosy"/>
</owl:Restriction>
```

Descriptions de classe

■ Property restriction

□ Cardinality constraints

Contrainte sur le nb de fois où une ressource est sujet d'une même propriété avec des valeurs différentes

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent"/>
  <owl:maxCardinality>2</owl:maxCardinality>
</owl:Restriction>
```

```
owl:minCardinality
owl:cardinality
```

Descriptions de classe

■ Intersection, Union, Complement

□ Intersection

```
<owl:Class rdf:ID="Man">
  <owl:intersectionOf
    rdf:parseType="Collection">
    <owl:Class rdf:ID="#Male"/>
    <owl:Class rdf:ID="#Person"/>
  </owl:intersectionOf>
</owl:Class>
```

□ Union owl:unionOf

Descriptions de classe

■ Intersection, Union, Complement

□ Complement : « négation »

```
<owl:Class rdf:ID="Man">
  <owl:complementOf rdf:resource="#Female"/>
</owl:Class>
```

Axiomes de classe

■ owl:Class

```
<owl:Class rdf:ID="Human"/>

<owl:Class rdf:ID="Man">
  <owl:complementOf rdf:resource="#Female"/>
</owl:Class>

<owl:Class rdf:ID="Man">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#hasParent"/>
    <owl:hasValue rdf:resource="#Sarkosy"/>
  </owl:Restriction>
</owl:Class>
```

Axiomes de classe

■ rdfs:subclassOf

```
<owl:Class rdf:ID="Opera">
  <rdfs:subclassOf rdf:resource="#MusicalWork"/>
</owl:Class>

<owl:Class rdf:about="#Opera">
  <rdfs:subclassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasLibrettist"/>
      <owl:minCardinality>1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subclassOf>
</owl:Class>
```

Axiomes de classe

■ owl:equivalentClass

- Equivalence en terme d'extension
- Les concepts ainsi définis restent différents

```
<owl:Class rdf:about="#FR_President">
  <owl:equivalentClass
    rdf:resource="#PrincipalResidentOfElysee"/>
</owl:Class>
```

Axiomes de classe

■ owl:disjointWith

```
<owl:Class rdf:about="#FR_President">
  <owl:equivalentClass
    rdf:resource="#PrincipalResidentOfElysee"/>
</owl:Class>
```

Axiomes de Propriété

■ Classes de propriétés OWL

- Sous-classes de rdf:Property
- owl:ObjectProperty
- owl:DataTypeProperty
- owl:AnnotationProperty
- owl:OntologyProperty

■ RDFS constructs

- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range

Axiomes de Propriété

■ Propriétés algébriques de propriétés

```
<owl:SymmetricProperty rdf:ID="hasSpouse"/>
<owl:TransitiveProperty rdf:ID="hasAncestor"/>
```

■ Relations entre propriétés

□ Relations inverse

```
<owl:ObjectProperty rdf:ID="hasChild">
  <owl:inverseOf rdf:resource="#hasParent"/>
</owl:ObjectProperty>
```

□ Relations équivalentes (en termes d'extension)

```
owl:equivalentProperty
```

■ Contraintes de cardinalité

```
<owl:FunctionalProperty rdf:ID="#hasMother"/>
<owl:InverseFunctionalProperty rdf:ID="#isMotherOf"/>
```

Individus

■ Descriptions des types et propriétés en RDF

■ Identités d'individus

□ owl:sameAs

```
<rdf:Description rdf:about="#William_Jefferson_Clinton">
  <owl:sameAs rdf:resource="#BillClinton"/>
</rdf:Description>
```

```
<rdf:Description rdf:ID="FootballTeam">
  <owl:sameAs rdf:resource="ns2:#SoccerTeam"/>
</rdf:Description>
```

- owl:differentFrom
- owl:allDifferent

Annotations

■ Sur les classes, propriétés et individus

- à l'aide de propriétés instances de la classe
owl:AnnotationProperty

- owl:versionInfo
- rdfs:label
- rdfs:comment
- rdfs:seeAlso
- rdfs:isDefinedBy

Entête d'ontologie

■ Une ontologie est une ressource

- qui peut donc être décrite avec des propriétés instances de la classe owl:OntologyProperty

```
<owl:Ontology rdf:about="">
  <owl:versionInfo>v 1.1 2008</owl:versionInfo>
  <rdfs:comment>An example ontology</rdfs:comment>
  <owl:imports
    rdf:resource="http://www.example.org/foo"/>
</owl:Ontology>
```

Versioning

- owl:versionInfo
- owl:priorVersion
- owl:backwardCompatibleWith
- owl:incompatibleWith
- owl:DeprecatedClass
- owl:DeprecatedProperty

OWL Full

- Mélange libre des constructs RDFS et OWL
 - owl:Class équivalent à rdfs:Class
 - owl:Thing équivalent à rdf:Resource
 - Toute ressource est un individu (y compris les classes et propriétés)
 - owl:ObjectProperty équivalent à rdf:Property
 - Et donc owl:DatatypeProperty subClassOf owl:ObjectProperty
- N'est pas décidable

OWL DL

- Pairwise separation between classes, datatypes, datatype properties, object properties, annotation properties, ontology properties, individuals, data values and the built-in vocabulary
 - Disjointness of the sets of object properties and datatype properties implies that 'inverse of', 'inverse functional', 'symmetric', and 'transitive' can never be specified for datatype properties
- No cardinality constraints can be placed on transitive properties or their inverses or any of their superproperties
- Most RDF(S) vocabulary cannot be used within OWL DL
- All classes and properties that one refers to are explicitly typed as OWL classes or properties, respectively
- Axioms (facts) about individual equality and difference must be about named individuals

OWL Lite

- OWL Lite forbids the use of:
 - owl:oneOf
 - owl:unionOf
 - owl:complementOf
 - owl:hasValue
 - owl:disjointWith
 - owl:DataRange
- N'autorise pas de cardinalité autre que 0 ou 1

Exercice

- Definition de classe en extension

```
<owl:Class rdf:ID="CouleurYeux">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:ID="Bleu"/>
    <owl:Thing rdf:ID="Vert"/>
    <owl:Thing rdf:ID="Marron"/>
  </owl:oneOf>
</owl:Class>
```

Exercice

- Intersection de classes

```
<owl:Class rdf:ID="Man">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="#Male"/>
    <owl:Class rdf:ID="#Person"/>
  </owl:intersectionOf>
</owl:Class>
```

Exercice

■ Classes complémentaires

```
<owl:Class rdf:ID="Male">
  <owl:complementOf rdf:resource="#Female"/>
</owl:Class/>
```

■ Classes disjointes

```
<owl:Class rdf:ID="Carre">
  <owl:disjointWith rdf:resource="#Rond"/>
</owl:Class>
```

Exercice

■ Restriction de toutes les valeurs d'une propriété

```
<owl:Class rdf:ID="Herbivore">
  <subClassOf rdf:resource="Animal">
    <subClassOf>
      <owl:Restriction>
        <owl:onProperty
          rdf:resource="#eats"/>
        <owl:allValuesFrom
          rdf:resource="#Plant"/>
      </owl:restriction>
    </subClassOf>
  </owl:Class>
```

Exercice

■ Restriction d'au moins une valeur de propriété

```
<owl:Class rdf:ID="Sportive">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hobby"/>
      <owl:someValuesFrom
        rdf:resource="#Sport"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Exercice

■ Imposer une valeur exacte

```
<owl:Class rdf:ID="DeuxRoues">
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="#nbRoues"/>
      <owl:hasValue>2</owl:hasValue>
    </owl:restriction>
  </subClassOf>
</owl:Class>
```

Exercice

■ Contraindre la cardinalité d'une propriété

```
<owl:Class rdf:ID="Person">
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="#nom"/>
      <owl:maxCardinality>1</owl:maxCardinality>
    </owl:Restriction>
  </subClassOf>
</owl:Class>
```