

RDF *Schema*

Cours « Recherche d'Information et Graphe de données »

Nicolas Delestre

Classe (rappel) 1 / 2

- En RDF il est possible d'associer un URI à une ou plusieurs classes :
 - en utilisant la propriété `rdf:type`
 - en utilisant le nom de la classe pour la balise décrivant la ressource en remplacement `rdf:Description` (sérialisation XML)
C'est ce que l'on avait vu avec les ensembles, séquences, etc. au cours précédent
 - en utilisant la propriété `a` (sérialisation turtle)

Classe (rappel) 2 / 2

Exemple (notation turtle)

```

@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sch:      <http://schema.org/> .
@prefix prov:    <http://www.w3.org/ns/prov#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix insarouen: <https://data.insa-rouen.fr/> .
@prefix iti:     <https://data.insa-rouen.fr/iti/> .
@prefix enseignant_iti: <https://data.insa-rouen.fr/iti/person/> .

insarouen:etablissement
  rdf:type sch:CollegeOrUniversity ;
  sch:name "INSA de Rouen Normandie"@fr .

enseignant_iti:NicolasDelestre
  rdf:type sch:Person ;
  sch:name "Nicolas Delestre" ;
  sch:jobTitle "enseignant"@fr ;
  sch:worksFor insarouen:etablissement ;
  sch:affiliation insarouen:etablissement ;
  sch:teaches cours_iti:prog_av .

cours_iti:rigraphe
  rdf:type sch:Course ;
  sch:name "Recherche d'information et Graphe de données"@fr ;
  sch:name "Information Retrieval and Data Graph"@en ;
  sch:provider insarouen:etablissement .

```

RDF Schema

RDFS

- RDFS est le langage de description des vocabulaires/schémas RDF
- Un vocabulaire RDF permet de définir les classes et les propriétés d'un schéma RDF
- Un vocabulaire RDF est décrit en RDF :
 - Un URI représente une classe s'il a la propriété `rdf:type` avec la valeur `rdfs:Class` (la métaclasse des classes)
 - la propriété `rdfs:subClassOf` permet de définir des héritages
 - Un URI représente une propriété s'il a la propriété `rdf:type` avec la valeur `rdf:Property` (la classe des propriétés)
 - la propriété `rdfs:domain` permet de définir le *domain* (si non défini alors toute classe est valide)
 - la propriété `rdfs:range` permet de définir le *range* (si non spécifié alors toute classe ou littéral est valide)
 - la propriété `rdfs:subPropertyOf` permet de définir des héritages
 - On peut utiliser les propriétés d'annotation : `rdfs:label`, `rdfs:comment`, `rdfs:isDefinedBy`, `rdfs:seeAlso`

Description de classes

```
schema:CollegeOrUniversity a rdfs:Class ;
  rdfs:label "CollegeOrUniversity" ;
  rdfs:comment "A college, university, or other third-level educational institution." ;
  rdfs:subClassOf schema:EducationalOrganization .

schema:EducationalOrganization a rdfs:Class ;
  rdfs:label "EducationalOrganization" ;
  rdfs:comment "An educational organization." ;
  rdfs:subClassOf schema:CivicStructure,
    schema:Organization .

schema:Organization a rdfs:Class ;
  rdfs:label "Organization" ;
  rdfs:comment "An organization such as a school, NGO, corporation, club, etc." ;
  rdfs:subClassOf schema:Thing ;
  owl:equivalentClass gs1:Organization,
    fibo-fnd-org-org:Organization .

schema:Thing a rdfs:Class ;
  rdfs:label "Thing" ;
  rdfs:comment "The most generic type of item." .
```



Description de propriétés

```
schema:name a rdf:Property ;
  rdfs:label "name" ;
  rdfs:comment "The name of the item." ;
  rdfs:subPropertyOf rdfs:label ;
  owl:equivalentProperty og:title,
    dct:title,
    cmns-txt:hasName ;
  schema:domainIncludes schema:Thing ;
  schema:rangeIncludes schema:Text .

schema:jobTitle a rdf:Property ;
  rdfs:label "jobTitle" ;
  rdfs:comment "The job title of the person (for example, Financial Manager).";
  schema:domainIncludes schema:Person ;
  schema:isPartOf <https://pending.schema.org> ;
  schema:rangeIncludes schema:DefinedTerm,
    schema:Text ;
  schema:source <https://github.com/schemaorg/schemaorg/issues/2192> .
```

Remarques

- Un triplé RDF n'instancie pas une propriété, une propriété n'est pas une classe
- Un triplé RDF utilise/référence une propriété
- La définition d'une propriété dans un schéma RDF est réalisé par instanciation de la classe `rdf:Property`
- `rdf:Property` est une classe, pas une métaclasse

Les (méta)classes pour créer des schémas

Principales métaclasses de RDFS

- `rdfs:Resource` est la métaclasse de toutes les ressources
- `rdfs:Class` est la métaclasse de toutes les classes
- `rdfs:Literal` est la métaclasse de tous les littéraux
- `rdfs:Container` est la métaclasse de tous les conteneurs (`rdf:Bag`, `rdf:Seq` et `rdf:Alt`)

Principales classes de RDF

- `rdf:Property` est la classe de toutes les propriétés
- `rdf:langString` est la classe de tous les littéraux avec une langue associée
- `rdf:XMLLiteral` est la classe de tous les littéraux exprimés en XML
- `rdf:HTML` est la classe de tous les littéraux exprimés en HTML



Les propriétés pour créer des schémas

Propriétés RDFS

Property name	domain	range
<code>rdfs:subClassOf</code>	<code>rdfs:Class</code>	<code>rdfs:Class</code>
<code>rdfs:subPropertyOf</code>	<code>rdf:Property</code>	<code>rdf:Property</code>
<code>rdfs:domain</code>	<code>rdf:Property</code>	<code>rdfs:Class</code>
<code>rdfs:range</code>	<code>rdf:Property</code>	<code>rdfs:Class</code>
<code>rdfs:label</code>	<code>rdfs:Resource</code>	<code>rdfs:Literal</code>
<code>rdfs:comment</code>	<code>rdfs:Resource</code>	<code>rdfs:Literal</code>
<code>rdfs:seeAlso</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdfs:isDefinedBy</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>

Les (méta)classes et propriétés RDF/RDFS sont donc décrites en RDF 1 / 3

rdfs:Resource

```
rdfs:Resource a rdfs:Class;  
  rdfs:comment "The class resource, everything.";  
  rdfs:isDefinedBy rdfs::  
  rdfs:label "Resource".
```

rdfs:Class

```
rdfs:Class a rdfs:Class;  
  rdfs:comment "The class of classes.";  
  rdfs:isDefinedBy rdfs::  
  rdfs:label "Class";  
  rdfs:subClassOf rdfs:Resource.
```

Les (méta)classes et propriétés RDF/RDFS sont donc décrites en RDF 2 / 3

rdf:Property

```
rdf:Property a rdfs:Class;
  rdfs:comment "The class of RDF properties.";
  rdfs:isDefinedBy rdf;;
  rdfs:label "Property";
  rdfs:subClassOf rdfs:Resource.
```

rdf:Literal

```
rdfs:Literal a rdfs:Class;
  rdfs:comment "The class of literal values, eg. textual strings and integers.";
  rdfs:isDefinedBy rdfs;;
  rdfs:label "Literal";
  rdfs:subClassOf rdfs:Resource.
```

Les (méta)classes et propriétés RDF/RDFS sont donc décrites en RDF 3 / 3

rdfs:domain

```
rdfs:domain a rdf:Property;  
  rdfs:comment "A domain of the subject property.";  
  rdfs:domain rdf:Property;  
  rdfs:isDefinedBy rdfs::  
  rdfs:label "domain";  
  rdfs:range rdfs:Class.
```

rdfs:range

```
rdfs:range a rdf:Property;  
  rdfs:comment "A range of the subject property.";  
  rdfs:domain rdf:Property;  
  rdfs:isDefinedBy rdfs::  
  rdfs:label "range";  
  rdfs:range rdfs:Class.
```

Un point de vue ensembliste des classes

Que signifie la propriété `rdfs:subClassOf` ?

- Si B est une sous-classe de A alors tous les éléments de B sont des éléments de A (la réciproque n'est pas forcément vraie)
- D'un point de vue ensembliste, B est un sous-ensemble de A ($B \subseteq A$)

Dans le contexte de `rdf:type`, `rdfs:domain` et `rdfs:range`

- Si une ressource x a pour type la classe B et que B est une sous-classe de A alors x a pour type la classe A
- Si une ressource x a pour propriété P et que P a pour domaine la classe A alors x a pour type la classe A
- Si une ressource x a pour propriété P avec comme valeur une ressource y et que P a pour range la classe A alors y a pour type la classe A



Une hiérarchie des propriétés

Que signifie la propriété `rdfs:subPropertyOf` ?

- Si p' est une sous-propriété de p
- Si p' est utilisé dans un triplé RDF p' alors :
 - p' peut être « remplacée » par p sans changer le sens de la déclaration, elle devient toutefois plus générique, moins précise
 - un nouveau triplé RDF peut être ajouté avec le même sujet, le même objet mais avec la propriété p

Les 14 règles RDFS (RDF 1.1 Semantics)

Règles RDFS

	Conditions	Triplés inférés
rdfs1	any IRI aaa in D	aaa rdf:type rdfs:Datatype .
rdfs2	aaa rdfs:domain xxx . yyy aaa zzz .	yyy rdf:type xxx .
rdfs3	aaa rdfs:range xxx . yyy aaa zzz .	zzz rdf:type xxx .
rdfs4a	xxx aaa yyy .	xxx rdf:type rdfs:Resource .
rdfs4b	xxx aaa yyy.	yyy rdf:type rdfs:Resource .
rdfs5	xxx rdfs:subPropertyOf yyy . yyy rdfs:subPropertyOf zzz .	xxx rdfs:subPropertyOf zzz .
rdfs6	xxx rdf:type rdf:Property .	xxx rdfs:subPropertyOf xxx .
rdfs7	aaa rdfs:subPropertyOf bbb . xxx aaa yyy .	xxx bbb yyy .
rdfs8	xxx rdf:type rdfs:Class .	xxx rdfs:subClassOf rdfs:Resource .
rdfs9	xxx rdfs:subClassOf yyy . zzz rdf:type xxx .	zzz rdf:type yyy .
rdfs10	xxx rdf:type rdfs:Class .	xxx rdfs:subClassOf xxx .
rdfs11	xxx rdfs:subClassOf yyy . yyy rdfs:subClassOf zzz .	xxx rdfs:subClassOf zzz .
rdfs12	xxx rdf:type rdfs:ContainerMembershipProperty .	xxx rdfs:subPropertyOf rdfs:member .
rdfs13	xxx rdf:type rdfs:Datatype .	xxx rdfs:subClassOf rdfs:Literal .



Les 2 règles RDF (RDF 1.1 Semantics)

Règles RDF

	Conditions	Triplé(s) inférés
rdfD1	xxx aaa "sss"^^ddd . for ddd in D	xxx aaa _:nnn . _:nnn rdf:type ddd .
rdfD2	xxx aaa yyy .	aaa rdf:type rdf:Property .

Définition

- Un raisonneur est un logiciel/une bibliothèque qui permet de faire de l'inférence à partir d'un graphe RDF
- L'objectif est de trouver au requêtage (SPARQL) des informations qui ne sont pas présentes au départ
- Plusieurs stratégies : compléter le graphe, améliorer le moteur SPARQL, augmenter la requête SPARQL

Compléter le graphe

- Utilisation de l'algorithme de « Chainage avant » : on parcourt la base de fait et tant qu'il y a une condition (partie gauches des règles) qui est satisfaite et que les triplés résultats ne sont pas présents, on les ajoute
- Avantages : requêtage simple, simple à implémenter
- Inconvénient : explosion du nombre de triplés

rdfs9

- Triplés présents :
 - `:Enseignant rdfs:subClassOf :Personne`
 - `:NicolasDelestre rdf:type :Enseignant`
- Triplé ajouté :
 - `:NicolasDelestre rdf:type :Personne`



Améliorer le moteur SPARQL

- Utilisation de l'algorithme de « Chainage arrière » :
 - on cherche les faits directs (pas de chainage)
 - si des parties droites des règles matchent avec la requête, on cherche aussi les parties gauches (récursivité)
- Avantage : on ne touche pas à la base de données
- Inconvénients : requêtage plus lent, algorithme plus complexe

rdfs9

- Si (une partie de) la requête correspond à :
`?x rdf:type ?C.`
- Alors on recherche aussi
`?SC rdfs:subClassOf ?C.`
`?x rdf:type ?SC.`

Augmenter la requête SPARQL

- Utilisation de l'opérateur UNION
 - si une partie de la requête correspond avec la partie droite d'une règle, on ajoute à la requête la partie gauche de la règle sous forme d'UNION

rdfs9

- Transformer la requête

```
SELECT ?x WHERE { ?x rdf:type :Personne }
```

- En

```
SELECT ?x WHERE {  
  ?x rdf:type :Personne  
  UNION  
  ?x rdf:type ?C .  
  ?C rdfs:subClassOf+ :Personne  
}
```

Conclusion

Ce que nous avons vu dans ce cours

- Pour créer des vocabulaires/schémas RDF, il faut utiliser les classes et propriétés de RDF/RDFS proposées par le W3C
- `schema.org` est un exemple de schéma RDF qui toutefois n'utilise pas `rdfs:domain` et `rdfs:range`
- Il est possible grâce à un raisonneur RDF de retrouver des informations qui n'étaient pas présentes au départ \Rightarrow Inférence

Ce qui nous reste à voir

- Comment s'assurer que les données publiées sont sémantiquement valides ?

SHACL

