Examen - web sémantique

Mars 2021

(durée : 1h00 – barême : 20 points. Les TP seront notés chacun sur 10. La note de l'examen aura un coefficient de 0,7 et celle des TP de 0,3)

1. **Questions de cours** (répondre de manière synthétique.. 4 phrases maximum par réponse) **(5pts)**

1.1 Les graphes de connaissance RDF et RDFS (2 pts)

1.1.1 Comment représenter en RDFS que C est une classe et i est une instance de C ?
C rdf:type owl:Class et i rdf:type C

1.1.2 Comment un raisonnement produit-il de nouvelles connaissances lorsqu'il existe des instances de classes reliées par la propriété rdfs:subClassOf ?
Si C1 rdfs:subClassOf C2, i1 rdf:type C1, ALORS

1.2 Utilisation des données liées ouvertes (3pts)

1.2.1 Comment peut-on utiliser l'URI représentant Paris de DBPedia (dbr:Paris) dans une application annotant un texte écrit en Russe à l'aide d'entités issues de DBPedia ? Mentionner 2 problèmes qui peuvent se poser.

1.2.2 Expliquez 2 manières dont un moteur de recherche peut utiliser les données liées ou les bases de connaissances disponibles sous forme de données liées.

2. – Exercice 2 (10 pts, 2 par question) : FUNCTIONAL et contraintes de cardinalité

Dans l'ontologie du cinéma sur laquelle vous avez travaillé en TP1, on trouve les classes Personne, Acteur et Genre, associées aux data type et object Properties suivants :

```
tp1:Personne a owl :Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :aGenre ;
                  owl:cardinality "1"^^xsd:nonNegativeInteger
                ] ;
tp1:Masculin a tp1:genre .
tp1:Féminin a tp1:genre .
tp1:Genre a owl :Class .
tp1:Film a owl:Class .
tp1:Artiste a tp1:Personne .

tp1:aGenre a owl:ObjectProperty ;
     rdfs:range tp1:Genre ;
     rdfs:domain tp1:Personne .
tp1:aRéalisé a owl:ObjectProperty ;
      rdfs:range tp1:Film ;
      rdfs:domain tp1:Personne ;
      owl:inverseOf :réaliséPar .
tp1:joueDansFilm a owl:ObjectProperty ;
         rdfs:range :Film ;
```

```
        rdfs:domain :Personne .

tp1:Homme a owl:Class ;
    owl:equivalentClass [ rdf:type owl:Class ;
                owl:intersectionOf ( tp1:Personne
                            [ a owl:Restriction ;
                              owl:onProperty tp1:aGenre ;
                              owl:hasValue tp1:Masculin
                            ]
                           )
                ] .

tp1:Acteur a owl:Class ;
     a tp1:Artiste ;
     rdfs:label "Actor"@en , "Acteur"@fr , "Comédien"@fr ;
     owl:equivalentClass [a  owl:Class ;
                 owl:intersectionOf ( tp1:Homme
                            [ a owl:Restriction ;
                              owl:onProperty tp1:joueDansFilm ;
                              owl:onClass tp1:Film ;
                                owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger
                            ]
                           )
                ] .
tp1:Realisateur a owl:Class ;
        owl:equivalentClass [ a owl:Class ;
                    owl:intersectionOf ( tp1:Personne
                            [ a owl:Restriction ;
                              owl:onProperty tp1:aRéalisé ;
                              owl:onClass tp1:Film ;
                              owl:minQualifiedCardinality 1"^^xsd:nonNegativeInteger
                            ]
                           )
                    ] .
```

2.1. Définir, en utilisant la syntaxe Turtle, trois entités tp1:François_Truffaut , tp1:JeanPierre_Leaud et tp1:400_coups de type owl:Thing ; représenter que tp1:François_Truffaut a réalisé tp1:400_coups et que tp1:JeanPierre_Leaud joue dans tp1:400_coups.

2.2. tp1:JeanPierre_Leaud sera-t-il classé comme tp1:Acteur ? pourquoi ?
tp1:François_Truffaut sera-t-il classé comme tp1:Realisateur ? pourquoi ?

2.3. On ajoute à la base de connaissances tp1 la propriété (dataProperty) tp1:durée-Film qui associe un tp1:Film et un entier, ainsi que deux sous-classes de tp1:Film : tp1:Court_metrage (film de durée inférieure à 60 mn) et tp1:Ling_metrage (film de durée supérieure à 60 mn). On ajoute le triplet tp1:400_coups tp1:durée-Film 99
Comment tp1:400_coups va-t-il être classé par le raisonneur ? Donner toutes les classes et expliquer pourquoi (donner les règles utilisées par le raisonneur).

FUNCTIONAL : le cours comporte une erreur sur sa définition. **Functional indique qu'une propriété relie une ressource à AU PLUS une autre ressource** (et non exactement 1).

2.4. Quelle est la différence entre le fait d'indiquer qu'une tp1:Personne a un unique lien vers un tp1:genre par la propriété tp1:aGenre comme dans l'encadré ci-dessus ET le fait de définir tp1:aGenre comme Functional ?

2.5. Si on associe 2 genres (tp1:Masculin et tp1:Feminin) à tp1:François_Truffaut, le raisonneur ne constate pas d'erreur à cause de l'hypothèse du monde ouvert. Il considère à partir de là que les deux entités tp1:Masculin et tp1:Feminin sont identiques. Quelles nouvelles classes seront inférées pour tp1:François_Truffaut ? quelle relation poser entre tp1:Masculin et tp1:Feminin pour éviter cela ?

3. Exercice 3 (5pts) : Utiliser COUNT et GROUP BY dans SPARQL

Dans la base de connaissances de la BNF, les trois éléments suivants contribuent à décrire des ouvrages présentant l'œuvre de la Jean de la Fontaine « la cigale et la fourmi »

```
<http://data.bnf.fr/11975188/jean_de_la_fontaine_la_cigale_et_la_fourmi/studies>
dcterms:subject <http://data.bnf.fr/ark:/12148/cb119751881#frbr:Work> .

<http://data.bnf.fr/ark:/12148/cb119751881> a skos:Concept ;
   bnf-onto:FRBNF 11975188 ;
   dcterms:created "1985-07-06" ;
   dcterms:modified "2016-01-27" ;
   rdfs:seeAlso <http://catalogue.bnf.fr/ark:/12148/cb119751881> ;
   skos:note "Fable appartenant au premier recueil, publié sous le titre \"Fables choisies mises
en vers par monsieur de La Fontaine\""@fr ;
   skos:prefLabel "La cigale et la fourmi"@fr ;
   foaf:focus <http://data.bnf.fr/ark:/12148/cb119751881#frbr:Work> .

<http://data.bnf.fr/ark:/12148/cb119751881#frbr:Work> a frbr:Work ;
   rdfs:label "La cigale et la fourmi" ;
   bnf-onto:firstYear 1668 ;
   bnf-onto:subject "Littératures" ;
   dcterms:creator <http://data.bnf.fr/ark:/12148/cb11910267w#foaf:Person> ;
   dcterms:date "1668" ;
   dcterms:description "Fable appartenant au premier recueil, publié sous le titre \"Fables
choisies mises en vers par monsieur de La Fontaine\""@fr ;
   dcterms:language <http://id.loc.gov/vocabulary/iso639-2/fre> ;
   dcterms:subject <http://dewey.info/class/800/> ;
   dcterms:title "La cigale et la fourmi"@fr ;
   rdagroup1elements:dateOfWork <http://data.bnf.fr/date/1668/> ;
   ore:isAggregatedBy <http://data.bnf.fr/ark:/12148/cb120083695#frbr:Work> ;
   = <http://data.bnf.fr/ark:/12148/cb119751881#about> ;
   foaf:depiction <http://gallica.bnf.fr/ark:/12148/bpt6k129255c.thumbnail>,
      <http://gallica.bnf.fr/ark:/12148/bpt6k54338578.thumbnail>,
      <http://gallica.bnf.fr/ark:/12148/bpt6k58388000.thumbnail>,
```

<https://upload.wikimedia.org/wikipedia/commons/6/6d/The_Ant_and_the_Grasshopper_by_Charles_H._Bennett.jpg> .

3.1 **Interprétation de ces définitions.** Le premier bloc définit un skos:concept qui renvoie à une notice de la BNF (une entrée de leur catalogue), et à laquelle on peut associer une œuvre qui va se matérialiser dans plusieurs ouvrages (qui ont chacun un éditeur et un aspect physique différent, mais parfois même une écriture différente, en vieux français ou français plus contemporain par exemple) (2 pts)

3.1.1 Quel est le type qui permet de représenter une œuvre dans le vocabulaire frbr ? quelles sont les deux propriétés utilisées pour donner son titre ? Quelle est l'objectProperty qui lie une entrée de catalogue et une œuvre ?

3.1.2 Sachant que la propriété dcterms:creator lie une œuvre à son auteur, quel est le concept qui, dans cette base de connaissances, représenterait Jean de la Fontaine ?

3.2 **Requêtes SPARQL sur ces données**. Rappel de la syntaxe de COUNT et GROUP BY pour compter le nombre d'entités en lien avec une autre entité : la requête suivante affiche, pour chaque film, le nombre d'acteurs connus dans la base tp1 comme jouant dans ce film. (3 pts)

```
SELECT ?film (COUNT (distinct ?acteur) as ?count)
    WHERE { ?acteur tp1:joueDansFilm ?film }
GROUP BY ?acteur
```

3.2.1 Ecrire une requête qui afficherait tous les identifiants et les titres des œuvres dont l'auteur est Jean de la Fontaine dans data.bnf.fr.

3.2.2 Ecrire une requête qui compte toutes les œuvres de tous les auteurs de la base (elle affiche, pour chaque auteur, son nom et le nombre de ses œuvres et, pour chaque oeuvre, son titre).