Université de Montpellier — Master Informatique

Traitement sémantique des données (HAI824I)

TP à rendre (OBDA avec Obi-wan)

A rendre à la fin de la séance :

- le fichier de nom `tp` avec vos réponses à chacune des questions (rendre le fichier au format pdf).
- le fichier `ontology.nt`
- le fichier `ris.json`

Et seulement ces fichiers !

Rappels :

- Lancement du serveur Obi-wan (ligne de commande) :
  `java -jar ../obiwan.jar server obi-wan.properties`
- Visualisation du système d'intégration :
  `http://localhost:8080/ris/sw-ris/index.html`
- Interrogation du système avec SPARQL :
  `http://localhost:8080/client/index.html`
- A chaque fois que vous modifiez les mappings (fichier `ris.json`) ou l'ontologie (fichier `ontology.nt`), vous devez tuer le serveur Obi-wan (Crtl C par exemple) et relancer le serveur pour que les changements soient pris en compte.

1 Complétion de l'ontologie

Ajoutez la classe `PodRacer` ("module de course") qui est une sous-classe de véhicule. Précisez également que la propriété *uses* a pour domaine Character (personnage).
Est-il nécessaire d'indiquer que les propriétés *pilotOf* et *usesWeapon* ont pour domaine `Character` (et pourquoi ?) ?

tp : Indiquez quels triplets vous ajoutez à l'ontologie (vous pouvez ne pas mettre les préfixes complets ici) et répondre à la question.

2 Ajout d'un mapping

Dans le fichier de mappings, ajoutez un mapping de nom "pilot2" similaire à "pilot" qui permet de récupérer également les objets de la classe PodRacer.

Ecrivez une requête SPARQL qui demande tous les x et z tels que x pilote quelque chose de type z *tel que* z *est un type particulier de véhicule (sous-classe de la classe des véhicules).*

Quelles sont les réponses à votre requête ?

tp : Donnez la requête SPARQL et les réponses - pas besoin de donner le mapping "pilot2".

3 Modification d'un mapping

Nous nous intéressons maintenant aux mappings depuis la BD `sw_imdb.db`. Modifiez le mapping de nom "title" et ajoutez de quoi récupérer aussi l'année de sortie du film via une propriété dont l'IRI est `<http://www.imdb.com/releaseYear>`.

Exécutez la requête suivante :

```
SELECT * WHERE {
  ?movie imdb:title ?title .
  ?movie imdb:releaseYear 1977 .
}
```

tp : Donnez le mapping et les réponses à la requête.

4 Lier les acteurs aux films

Créez un mapping de nom "actress" qui, à partir de la table `casting` de `sw_imdb.db`, extrait les actrices et les films dans lesquels elles jouent. Les triplets produits utilisent une propriété dont l'IRI est `<http://www.imdb.com/actressIn>`.

Répondez à la requête suivante ("quelle actrice joue dans un film ? ") :

```
SELECT ?name ?title WHERE {
  ?actress imdb:fullName ?name .
  ?actress imdb:actressIn ?movie .
}
```

Faites un mapping similaire pour les acteurs masculins. Les triplets produits utilisent une propriété dont l'IRI est `<http://www.imdb.com/maleActorIn>`.

Répondez à la requête suivante :

```
SELECT ?name ?title WHERE {
  ?actor imdb:fullName ?name .
  ?actor imdb:maleActorIn ?movie .
}
```

tp : Donnez les deux mappings et les réponses obtenues au deux requêtes SPARQL.

5 Généralisation de propriétés

Ajouter dans l'ontologie une propriété `imdb:actorIn` qui généralise les deux propriétés `imdb:actressIn` and `imdb:maleActorIn` (attention, il vous faut indiquer le préfixe en entier dans l'ontologie).

Vérifiez par une requête SPARQL que la nouvelle propriété se comporte bien comme une généralisation des 2 propriétés précédentes.

tp : Donnez les triplets ajoutés à l'ontologie (sans mettre les préfixes complets ici), la requête SPARQL et le nombre de réponses obtenues.

6 Lier les acteurs aux personnages

Ajoutez un mapping à partir de la table `casting` de `sw_imdb.db` qui permette de lier (un identifiant d') acteur - homme ou femme - au personnage qu'il ou elle joue via la propriété `imdb:plays`.

Indication : Si besoin, vous pouvez utiliser dans la requête SQL la syntaxe suivante : `category in ('actor', 'actress')`. Ceci évite d'avoir à créer deux mappings.

Exécutez la requête ci-dessous :

```
SELECT ?actorName ?character WHERE
{ ?actor imdb:fullName ?actorName .
?actor imdb:plays ?character .
?character sw:uses ?object .
?object rdf:type sw:StarShip .
}
```

tp : Donnez le mapping ajouté et les réponses à la requête.

7 Lier les personnages aux films

On veut retrouver quels personnages apparaissent dans quels films. Peut-on le faire avec la requête suivante :

```
SELECT  ?character ?movieTitle
WHERE {
  ?actor imdb:plays ?character .
  ?actor imdb:actorIn ?movie .
  ?movie imdb:title ?movieTitle .
 }
```

1. dans le cas de *cette* base de données ?
2. dans le cas d'une base de données où il peut y avoir des informations sur des films quelconques ?

Pourquoi ?
Proposez une façon de faire.

Précision de syntaxe des mappings. Si votre solution implique un mapping dans lequel il y a une jointure SQL à faire (donc plusieurs tables), il faut préfixer le nom de chaque attribut par le nom de sa table (T.A pour un attribut A d'une table T) dans la requête SQL et dans les templates qui utilisent ces attributs.

tp : Répondre à la question et donnez votre solution (expliquez bien les différents éléments de votre solution).

8 Finalisation de l'ontologie

Insérez la classe `imdb: Movie` dans votre ontologie, de façon à ce qu'elle soit correctement peuplée par les (identifiants de) films de la BD. Il s'agit de faire apparaitre cette classe dans l'ontologie, vous n'avez pas à ajouter de mappings.

Vérifiez par une requête SPARQL que vous avez effectivement des instances de la classe Movie (combien ?)

Voyez-vous d'autres modifications à apporter pour améliorer votre ontologie ?

tp : Indiquez quelles modifications vous avez apportées à l'ontologie pour intégrer `imdb: Movie`*, la requête SPARQL de test et le nombre de réponses obtenues. Indiquez ensuite si vous voyez d'autres modifications à apporter pour améliorer votre ontologie.*

9 Sous le capot

Considérons cette requête SPARQL très simple :

```
SELECT ?name WHERE
{ ?actor imdb:fullName ?name .
  ?actor imdb:actorIn ?movie .
  ?movie imdb:title "Star Wars: Episode IV - A New Hope" .
}
```

Nous avons choisi la stratégie d'interrogation qui reformule la requête avec l'ontologie, puis réécrit les reformulations avec les mappings (option "REW-CA" dans le fichier obi-wan.properties).

Combien y-a-t-il de reformulations de cette requête avec l'ontologie, puis de réécritures avec les mappings ? Comment ces reformulations et réécritures sont-elles obtenues ?

Vous pouvez vous aider du répertoire "sessions" (et sa visualisation). Remarquez au passage qu'Obi-wan associe un plan d'exécution à la réécriture obtenue, puis optimise ce plan, et finalement l'implémente avec des opérations précises (par exemple "join" est remplacé par un algorithme de "join" particulier).

tp : Expliquez comment se fait la reformulation puis la réécriture de la requête. On doit pouvoir comprendre pourquoi il y a ces reformulations et ces réécritures.