INF6410 - Ontologies et web sémantique
Contrôle périodique - Automne 2016
Professeur: Michel Gagnon
École Polytechnique de Montréal

21 novembre 2016

1 RDF (5 points)

a) (3 points) Soit la situation suivante :

> Les intervenants du Centre d'amitié autochtone de Val-d'Or (CAAVD) se sont dits profondément déçus, jeudi, d'apprendre qu'aucun des policiers de Val-d'Or ayant été ciblés par des plaintes de femmes autochtones ne sera accusé.

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix ex: <http://www.exemple.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

ex:CentreAmitieAutochtone ex:localisation dbpedia:Val_Dor ;
   rdf:label "Centre d'amitié autochtone de Val D'or" ;
   skos:altLabel "CAAVD" .

[] a rdf:Bag ;
   ex:provenance ex:CentreAmitieAutochtone ;
   ex:typeDesMembres ex:Intervenant ;
   ex:informéDe ex:g1 ;acÙÙFÙ       ex:g1 a ex:Nouvelle ;
         ex:date "2016-11-17"^^xsd:Date .

ex:g1 {
   []  a owl:NegativePropertyAssertion ;
   owl:assertionProperty  ex:accusation ;
   owl:targetIndividual
       [ a ex:Policier ;
         ex:cibleDe
            [a rdf:Plainte ;
             ex:source
                [ a ex:Femme ;
                  a ex:Autochtone ] ] ] .
}
```

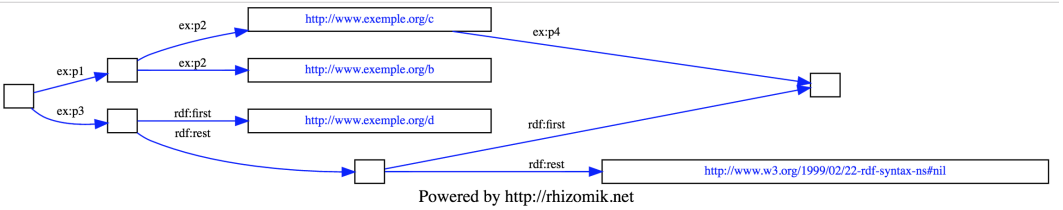
Représentez cette situation en RDF, en utilisant la méthode de graphe nommé (dataset). Assurez-vous que tous les éléments importants soient correctement représentés. Vous noterez aussi la présence d'une négation, ce qui vous obligeraà contourner les limites de RDF, qui ne permet pas la négation. Par contre, vous savez qu'il est possible d'exprimer la négation en OWL.

b) (1 point) Dites quelle sémantique nous devrions utiliser pour interpréter le graphe de la question précédente et justifiez votre choix.

L'interprétation où chaque graphe définit son propre contexte (c'est sa dénotation qui nous intéresse)

c) (1 point) Dessinez le graphe RDF suivant :

```
@prefix ex: <http://www.exemple.org/> .
[ex:p1 [ ex:p2 ex:b, ex:c ];
 ex:p3 (ex:d _:n1) ] .
ex:c ex:p4 _:n1 .
```



2 Sémantique de RDF (3 points)

Soient les deux graphes RDF suivants :
G1 :

```
@prefix local: <http://www.polymtl.ca/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdfsyntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

local:a lrdf:type local:B.
local:c local:p [] .
local:p rdfs:domain local:C .
```

G2 :

```
@prefix local: <http://www.polymtl.ca/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdfsyntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

local:a rdf:type local:B.
local:B rdfs:subClassOf local:C .
```

a) (1 point) Dites lequel des énoncés suivants est vrai (justifiez votre réponse)

A. G1 et G2 sont équivalents

B. G1 implique G2

C. G2 implique G1

D. Aucune relation logique entre G1 et G2

D. Aucune relation logique entre ces graphes. G1 n'implique pas G2, parce que rien dans G1 n'implique de B soit sous-classe de C. G2 n'implique pas G1, parce que rien dans G2 n'implique les deuxième et troisième triplets de G1.

b) (1 point) Proposez une interprétation qui est vraie pour les deux graphes.

$I_s : a \rightarrow \mathbf{a}, b \rightarrow \mathbf{b}, c \rightarrow \mathbf{c}, p \rightarrow \mathbf{p}, B \rightarrow \mathbf{B}, C \rightarrow \mathbf{C},$
$\quad subClassOf \rightarrow \mathbf{SC}, type \rightarrow \mathbf{T}, domain \rightarrow \mathbf{D}$
$I_{EXT}(\mathbf{P}) = \{\langle \mathbf{c}, \mathbf{a} \rangle, \ldots\}$
$I_{EXT}(\mathbf{T}) = \{\langle \mathbf{a}, \mathbf{B} \rangle, \langle \mathbf{a}, \mathbf{C} \rangle, \langle \mathbf{c}, \mathbf{C} \rangle \ldots\}$
$I_{EXT}(\mathbf{D}) = \{\langle \mathbf{P}, \mathbf{C} \rangle, \ldots\}$

$I_{EXT}(\mathbf{SC}) = \{\langle \mathbf{B}, \mathbf{C} \rangle, \ldots\}$
$IC_{EXT}(\mathbf{B}) = \{\mathbf{a}, \ldots\}$
$IC_{EXT}(\mathbf{C}) = \{\mathbf{a}, \mathbf{c}, \ldots\}$

c) (1 point) Jean et Marie ont tous les deux suivi le cours INF6410, mais il sont en désaccord sur la sémantique de RDF. Jean dit que pour n'importe quelle paire de graphes G_1 et G_2, on peut toujours trouver une interprétation qui est valide pour les deux graphes. Marie lui prouve que ce n'est pas vrai, en lui montrant deux graphes pour lesquels il est impossible de trouver une interprétation commune. Proposez un exemple que Marie a pu utiliser pour convaincre Jean.

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix ex: <http://www.exemple.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

ex:G1 {
   ex:a ex:p ex:y
}

 ex:G2 {
   ex:a rdf:type ex:p
}
```

Dans un cas, `ex:p` est une propriété, et dans l'autre, une classe. Les classes et les propriétés sont des sous-ensemble disjoints de IR. Il est donc impossible que `ex:p` ait la même interprétation pour les deux graphes.

3 Logique descriptive (5 points)

a) (1,5 point) Soit la base de connaissances suivante :

Chien $\equiv$ Vivant $\sqcap \exists$aime.Humain
$\exists$aime.$\top \sqsubseteq$ Animal

Prouvez de manière formelle, par la méthode des tableaux, que Chien $\sqsubseteq$ Animal.

Chien $\equiv$ Vivant $\sqcap \exists$aime.Humain
$\top \sqsubset \forall$aime.$\bot \sqcup$ Animal

$(\text{Chien} \sqcap \neg\text{Animal})(x)$
$\text{Chien}(x)$
$\neg\text{Animal}(x)$
$(\text{Vivant} \sqcap \exists\text{aime.Humain})(x)$
$\text{Vivant}(x)$
$(\exists\text{aime.Humain})(x)$
$\text{aime}(x, y)$
$\text{Humain}(y)$
$(\forall\text{aime.}\bot \sqcup \text{Animal})(x)$

$(\forall\text{aime.}\bot)(x)$ | $\text{Animal}(x)$
$\bot(y)$ | $\square$
$\square$

b) (0,5 point) Exprimez dans vos mots, de manière claire et concise, ce que veut dire l'axiome suivant en logique descriptive : $\exists$aime.$\top \sqsubseteq$ Humain $\sqcap \exists$aime.Humain

Tous ceux qui aiment sont des Humain qui aiment au moins un humain.

c) (2 points) Écrivez en OWL (en utilisant la notation Turtle) une ontologie qui correspond à la description suivante :

> Un médicament est soit disponible librement au comptoir, soit seulement avec une prescription. Tout médicament contient une substance active et peut être substitué par un autre médicament. Le Panadol, dont la substance active est le paracétamol, est en vente libre et peut être substitué par le Tylénol. Le Tramadol ne peut être vendu qu'avec une prescription.

Assurez-vous de définir le plus possible vos classes (restrictions nécessaires, classes disjointes, etc.) et vos propriétés (domaine, image, propriétés fonctionnelles, propriétés transitives, etc.).

```
@prefix ex: <http://exemple.org/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://exemple.org/> rdf:type owl:Ontology .

ex:peutEtreSubstituéPar
    a owl:ObjectProperty ,
    owl:SymmetricProperty ,
    owl:TransitiveProperty ,
    owl:IrreflexiveProperty .

ex:substanceActive
    a owl:ObjectProperty ;
    rdfs:domain ex:Médicament ;
    rdfs:range ex:Substance .

ex:precriptionRequise
  a owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain ex:Médicament ;
  rdfs:range xsd:boolean .

ex:MedicamentEnVenteLibre
    a owl:Class ;
    owl:equivalentClass
      [owl:intersectionOf
       ( ex:Médicament
         [ a owl:Restriction ;
           owl:onProperty ex:precriptionRequise ;
           owl:hasValue false ] ) ] .

ex:MedicamentSousPrescription
    a owl:Class ;
    owl:equivalentClass
      [owl:intersectionOf
       ( ex:Médicament
         [ a owl:Restriction ;
           owl:onProperty ex:precriptionRequise ;
```

```
           owl:hasValue true ] ) ] .

ex:Substance rdf:type owl:Class .

ex:Médicament
    a owl:Class ;
    rdfs:subClassOf
       [a owl:Restriction ;
        owl:onProperty ex:substanceActive ;
        owl:someValuesFrom ex:Substance] ,
       [a owl:Restriction ;
        owl:onProperty ex:peutEtreSubstituéPar ;
        owl:allValuesFrom ex:Médicament ] .

ex:Panadol
     a owl:NamedIndividual , ex:MedicamentEnVenteLibre ;
     ex:peutEtreSubstituéPar ex:Tylénol ;
     ex:substanceActive ex:Paracétamol .

ex:Paracétamol
     a owl:NamedIndividual , ex:Substance .

ex:Tramadol
     a owl:NamedIndividual , ex:MedicamentSousPrescription .

ex:Tylénol
     a owl:NamedIndividual , ex:MedicamentEnVenteLibre ;
     ex:substanceActive ex:Paracétamol .
```

d) (1 point) Soit l'ontologie de la question précédente. On aimerait spécifier qu'un médicament ne peut être substitué que par un autre qui contient la même substance active. Peut-on représenter cela en OWL ? Justifiez votre réponse et montrez comment on peut représenter cette situation.

Non. Il faudrait représenter cela par une restriction qui établirait un lien d'identité entre la substance active du médicament et la substance active du médicament qui est objet de la restriction. Or, on ne peut pas faire ça en logique descriptive. Il faudrait don utiliser une règle de la forme suivante :

Médicament(x) $\wedge$ substanceActive(x,y) $\wedge$ peutEtreSubstituépar(x,z) $\rightarrow$ substanceActive(z,y)

4 SPARQL (3 points)

Soit la base de connaissances suivante :

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

  _:p1 foaf:name "Paul" ;
       foaf:age "17"^^xsd:nonNegativeInteger ;
       foaf:mbox "paul@abc.com" .
  _:p2 foaf:name "Ana" ;
       foaf:age "27"^^xsd:nonNegativeInteger ;
       foaf:mbox "ana@abc.com" .
  _:p3 foaf:name "Léa" ;
       foaf:age "21"^^xsd:nonNegativeInteger ;
       foaf:mbox "lea@uuu.com" .
  _:p4 foaf:name "Guy" ;
       foaf:age "19"^^xsd:nonNegativeInteger ;
       foaf:mbox "guy@uuu.com" .
  _:p5 foaf:name "Gil"  ;
       foaf:age "20"^^xsd:nonNegativeInteger ;
       foaf:mbox "gil@xyz.ca" .
  _:p6 foaf:name "Lola" ;
       foaf:age "30"^^xsd:nonNegativeInteger ;
       foaf:mbox "lola@xyz.ca" .

  _:p1 foaf:knows _:p2 .
  _:p1 foaf:knows _:p3 .
  _:p2 foaf:knows _:p1 .
  _:p2 foaf:knows _:p3 .
```

a) (1,5 point) Dites quel sera le résultat de chacune des requêtes suivantes :

Requête Q1 :

```
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>

  SELECT DISTINCT ?n
  WHERE {
     ?x foaf:mbox ?m ;
        foaf:name ?n ;
        foaf:age ?a1 ;
        foaf:knows/foaf:age ?a2
    FILTER (?a2 > ?a1)
  }
```

	n
1	"Paul"

Requête Q2 :

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?dom (AVG(?age) as ?a)
WHERE {
   ?x foaf:age ?age ;
      foaf:name ?n ;
      foaf:mbox ?m .

   BIND (strafter(?m,'@') as ?dom)
}
GROUP BY ?dom
```

Remarque : la fonction *strafter(s,p)* retourne la sous-chaîne qui commence immédiatemment après la sous-chaîne *p*.

	dom	a
1	"abc.com"	22.0
2	"uuu.com"	20.0
3	"xyz.ca"	25.0

Requête Q3 :

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT DISTINCT ?n
WHERE {
  {
    SELECT DISTINCT ?p
     WHERE {
       ?p foaf:age ?a
       FILTER (?a > 20)
     }
  }
  ?x foaf:name ?n .
  {?x foaf:knows ?p } UNION {?p foaf:knows ?x }
}
```

	n
1	"Paul"
2	"Ana"
3	"Léa"

b) (0,5 point) Écrivez une requête SPARQL qui permet de déterminer si le graphe des connaissances contient un cycle (c'est le cas dans notre exemple, puisque Paul connaît Ana, qui elle-même connaît Paul).

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

ASK
WHERE {
   ?x foaf:knows+ ?x
}
```

c) (0,5 point) Écrivez une requête SPARQL pour extraire le nom des personnes qui ont plus de 18 ans et moins de 30 ans, qui connaissent quelqu'un, et dont le courriel n'est pas dans le domaine "uuu.com".

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?name
WHERE {
   ?x foaf:name ?name ;
      foaf:age ?age ;
      foaf:mbox ?mbox ;
      foaf:knows [] .
FILTER ( ?age > 18 && ?age < 30 && !contains(?mbox,"uuu.com"))
}
```

ou

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
  SELECT DISTINCT ?name
  WHERE {
     {
      ?x foaf:name ?name ;
        foaf:age ?age ;
        foaf:knows [] .
      FILTER ( ?age > 18 && ?age < 30)
     }
     MINUS
     {
       ?x foaf:mbox ?mbox ;
       FILTER (contains(?mbox,"uuu.com"))
     }
}
```

d) (0,5 point) Expliquez, en fournissant un exemple, pourquoi l'ordre des items dans une requête SPARQL peut influencer les performances lors de son exécution.

Supposons la requête SPARQL suivante :

```
  SELECT DISTINCT ?x ?z
  WHERE {
     {
      ?x p ?y .
      ?y q ?z .
    }
```

Si le nombre de de triplets utilisant la propriété p est beaucoup plus élevé que pour la propriété q, cette requête va beaucoup plus souvent instancier inutilement le premier pattern que si on avait utilisé l'autre ordonnancement pour les deux triplets.

5 Question à faire à la maison (4 points)

Répondez à une seule des deux questions suivantes :

a) Reprenez l'exemple du réfrigérateur intelligent que nous avons fait en classe. Vous aimeriez que votre base de conaissances soit enrichies par des données du Web sémantique. Vous hésitez entre DBpedia et Wikidata. Proposez une réflexion permettant de choisir entre ces deux ressources. Pour Wikidata, déterminez laquelle des implémentations RDF serait la plus adéquate (fournissez des justifications).

b) Reprenez l'exemple du réfrigérateur intelligent que nous avons fait en classe. On aimerait que les requêtes à la base de connaissances se fassent en langue naturelle. Expliquez comment cela pourrait être fait et proposez une réflexion sur la possibilité d'utiliser une ontologie du domaine pour faciliter la tâche.