Nom :
Prénom :

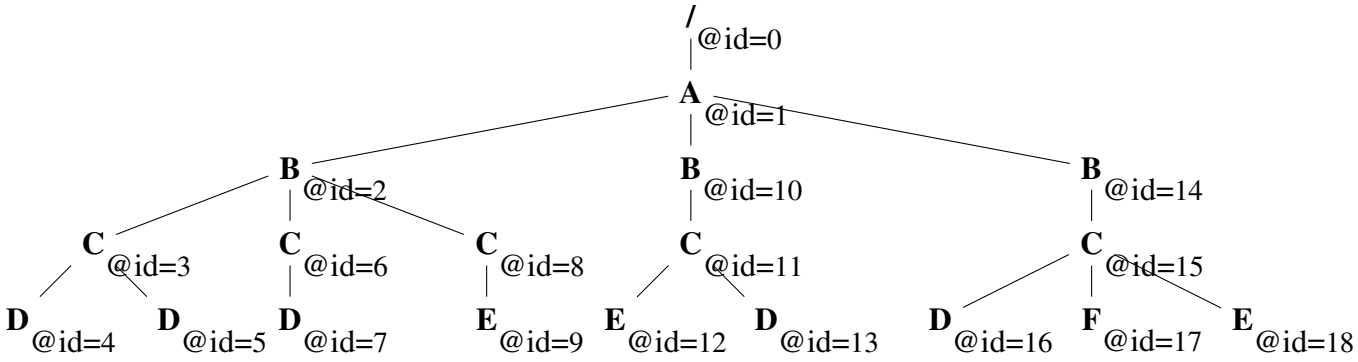
ASWS Apprentissage Symbolique et Web Sémantique

Examen du 21 novembre 2018 – partie Web Sémantique – documents autorisés

Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 20 points (12 questions) n'a qu'une valeur indicative.

XML/Xpath

Soit donné l'arbre XML suivant où chaque noeud est identifié par un attribut `@id` qui correspond à sa position dans l'ordre préfixe de l'arbre. L'identifiant de la racine est égal à 0, son élément fils de type `A` a l'identifiant 1 etc.:



Le résultat d'une expression XPath est une liste de noeuds DOM triés dans l'ordre du document et sans doublons. Par exemple, l'expression XPath `/descendant::B/@id` retourne les identifiants 2, 10, 14.

Question 1

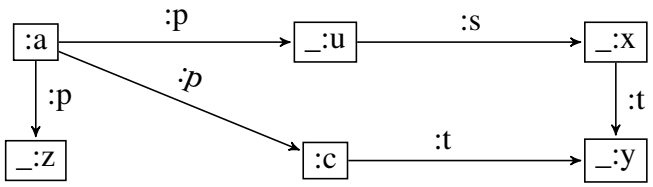
Donnez les expressions XPath suivantes.

1. Tous les noeuds C qui ont au moins un enfant E et un enfant D (11, 15).
2. Tous les noeuds B qui n'ont pas de descendant F (2, 10).
3. Tous les noeuds avec au moins trois enfants (1,2,15).
4. La dernière feuille de l'arbre (18).

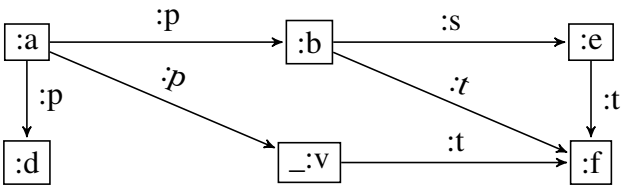
RDF

Soit les deux graphes RDF suivants:

G1:



G2:



On suppose que les noeuds et les propriétés appartiennent à l'espace de nom `http://exemple.org/`. Chaque noeud est étiqueté par son identifiant ou par une lettre précédée par '_:' s'il s'agit d'un noeud blanc. Les arcs sont étiquetés par les types des propriétés. Aucune propriété est transitive.

Question 2

Dessinez ou donnez en format Turtle la **forme normale** de $G2$ (sans déclaration des espaces de nom).

Question 3

Est-ce que $G1 \models G2$ et/ou $G2 \models G1$? Entourez la bonne réponse et justifiez la formellement.

Schémas/règles RDFS:

Voici un sous-ensemble des règles RDF/S vues en cours:

- rdfD2: a p b . ⇒ p rdf:type rdf:Property .
- rdfs2: p rdfs:domain x . a p b . ⇒ a rdf:type x .
- rdfs3: p rdfs:range x . a p b . ⇒ b rdf:type x .
- rdfs4b: a p b . ⇒ b rdf:type rdf:Resource .
- rdfs5 : p rdfs:subPropertyOf q . q rdfs:subPropertyOf r . ⇒ p rdfs:subPropertyOf r .
- rdfs7: p rdfs:subPropertyOf q . a p b . ⇒ a q b .
- rdfs9: c rdfs:subClassOf d . a rdf:type c . ⇒ a rdf:type d .
- rdfs11: c rdfs:subClassOf d . d rdfs:subClassOf e . ⇒ c rdfs:subClassOf e .

Soit le schéma RDFS *schema.rdfs* suivant :

```
:B rdfs:subClassOf :A .
:C rdfs:subClassOf :A .
:E rdfs:subClassOf :F .

:p rdfs:domain :B ; rdfs:range :B .
:q rdfs:domain :C ; rdfs:range :B .
:s rdfs:domain :E ; rdfs:range :C .
:s rdfs:subPropertyOf :q .
```

Question 4

Donnez pour chaque expression RDF suivant *les classes* des ressources `:x`, `:y` et `:z` qui sont *inférées par les règles RDF/S* (RDF/S entailment).

- Graphe 1:

```
:x :p :x .
```

- Graphe 2:

```
:x :q :y .
:x :s :y .
```

- Graphe 3:

```
:x :p :y .
:y :q :z .
:z :s :x .
```

SPARQL

Soit un graphe RDF/S *ratp.ttl*:

```
:Metro rdf:subclassOf :Ligne .
:Bus rdf:subclassOf :Ligne .
:RER rdf:subclassOf :Ligne .
:arret rdfs:domain :Ligne ; rdfs:range :Station .

:rer_b rdf:type :RER; :arret :st_michel, :luxembourg , :denfert .
:bus_14 rdf:type :Bus; :arret :port_royal , :denfert .
:bus_15 rdf:type :Bus; :arret :denfert .
:m4 rdf:type :Metro; :arret :raspail, :denfert .
:m6 rdf:type :Metro; :arret :raspail, :denfert, :port_royal .
:m5 rdf:type :Metro; :arret :st_michel, :bastille, :republique .
```

Question 5

Exprimez les requêtes suivantes en SPARQL sur *ratp.ttl*. On suppose que le graphe est saturé avant d'être interrogé et vous n'êtes pas obligé de spécifier les espaces de nom utilisés dans les requêtes.

1. Les stations où se croisent au moins deux lignes de transport : `:st_michel`, `:denfert`, `:port_royal`, `:raspail`.
2. Les bus qui s'arrêtent à Denfert mais pas à Port Royal: `:bus_15`
3. Toutes les stations qu'on peut atteindre de Port Royal avec un changement au maximum: `:raspail`, `:denfert`, `:st_michel`, `:luxembourg`, `:port_royal`

Question 6

Expliquez en français le résultat de la requête ci-dessous:

```
select distinct ?m
from <ratp.ttl>
where { { ?m a :Metro }
        minus { { :bus_14 :arret ?s . ?m a :Metro . }
                minus { { ?m :arret ?s } } } }
```

Question 7

Est-ce que la requête suivante retourne le même résultat que la requête précédente? Justifiez votre réponse.

```
select distinct ?m
from <ratp.ttl>
where { { ?m a :Metro . }
        minus { { :bus_14 :arret ?s . }
                minus { { ?m :arret ?s } } } }
```

Requêtes et Raisonnement

On donne l'ontologie *RATP* suivante exprimée en Logique de Description:

```
:ligne owl:inverseOf :arret .
:StationMetro owl:subClassOf [ a owl:Restriction ; owl:onProperty :ligne ;
                               owl:someValuesFrom :Metro ] .
:StationRER owl:subClassOf [ a owl:Restriction ; owl:onProperty :ligne ;
                             owl:someValuesFrom :RER ] .
:StationFerre owl:unionfOf (:StationMetro :StationRER) .
```

Question 8

Définissez un ensemble de *règles forward* (syntaxe Jena) qui génèrent les triplets qu'on peut inférer par l'ontologie *RATP*.

Question 9

Traduisez les requêtes SPARQL suivante en une nouvelle requête SPARQL qui calcule la même réponse **sans** utiliser la propriété *:ligne* et les concepts *:StationMetro*, *:StationRER* et *:StationFerre*

- ```
  select ?l where { ?s :ligne ?l . }
  ```
- ```
  select ?l
  where { ?s rdf:type :StationMetro;  :ligne ?l . }
  ```
- ```
  select ?l
  where { ?s rdf:type :StationFerre;  :ligne ?l . }
  ```

Plans de Jointure et Optimisation

On veut exécuter la requête SPARQL suivante:

```
select distinct ?e ?a ?b ?h ?u from <ratp.ttl>
where {
t1        ?a :arret ?u .
t2        ?a rdf:type :Metro
t3        ?b rdf:type :Metro
t4        ?b :arret ?v .
t5        ?c rdf:type :Metro
t6        ?c :arret :denfert .
t7        ?u :ligne ?b .
t8        ?v :ligne ?c . }
```

Question 10

Dessinez le *graphe de jointure* et le *graphe de variables* de la requête précédente. Vous pouvez numéroter les triplets dans l'ordre de leur apparition dans la requête ($t1$, $t2$, ..., $t8$).

Question 11

Donnez les ensembles indépendants *maximaux* du graphe de variables et un plan de jointure n-aires efficace correspondant.

Question 12

Donnez un plan linéaire efficace et expliquez pourquoi il est meilleur que le plan (((((((t1, t2), t3), t4), t5), t6), t7), t8).
```