

ASWS Apprentissage Symbolique et Web Sémantique – CORRIGÉ

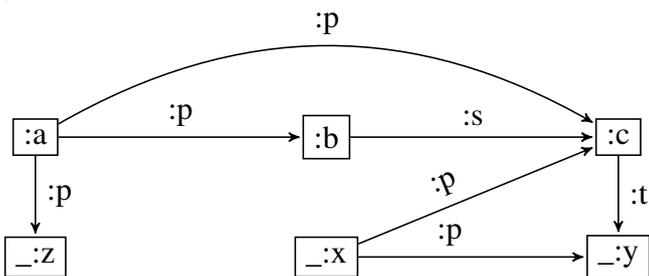
Examen du 22 novembre 2017 – partie Web Sémantique – documents autorisés

Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 20 points (10 questions) n'a qu'une valeur indicative.

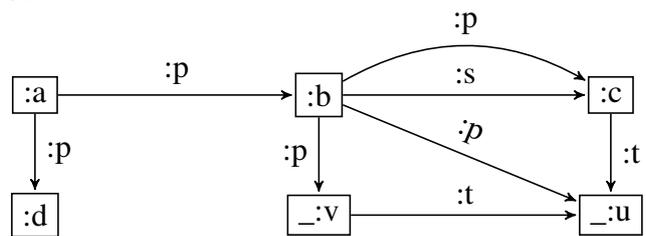
1 RDF (5 pts)

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 et **la propriété :p est transitive**.

Question 1 (1 point)

Est-ce que la définition Turtle ci-dessous est équivalente au graphe G1 (oui/non) ?

```
:a :p :b, :c, []. :b :s :c. [] :p :c, _:y. :c :t _:y.
```

Réponse :

Solution: oui

Question 2 (2 points)

Donnez les formes normales de G1 et de G2 en format Turtle (vous n'êtes pas obligé de déclarer les espaces de nom).

Réponse :

Solution:

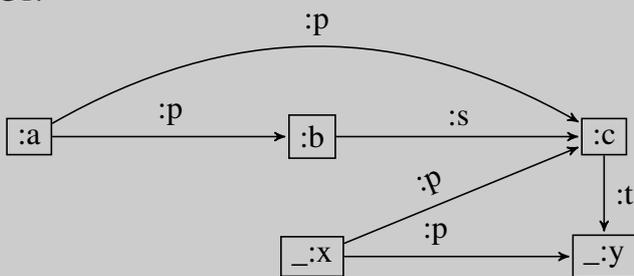
G1:

```
:a :p :b, :c .
:b :s :c .
[] :p :c, _:y .
:c :t _:y .
```

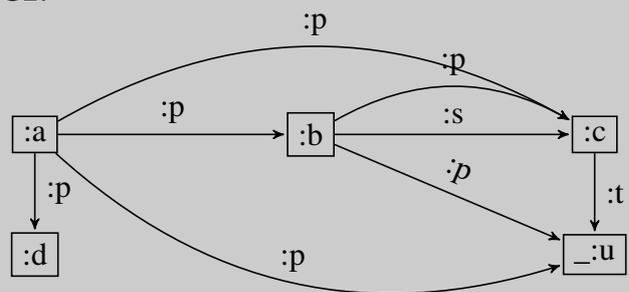
G2:

```
:a :p :b, :c, :d, _:u .
:b :s :c; :p :c, _:u .
:c :t _:u .
```

G1:



G2:



Question 3 (2 points)

Est-ce que $G1 \models G2$ et/ou $G2 \models G1$? Justifiez vos réponses formellement.

Réponse :

- $G1 \models G2$?

- $G2 \models G1$?

Solution:

- $G1 \not\models G2$: NON; il n'existe pas de mapping m tel que $m(G2)$ est un sous-graphe de $G1$ (l'arc $(:b :p :c)$ n'existe pas dans $G1$).
- $G2 \models G1$: OUI; avec le mapping $m : _ : x \rightarrow b, _ : z \rightarrow d, m(G1)$ est un sous-graphe de $G2$.

2 SPARQL (6 pts)

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

```

:memberOf    rdfs:domain :Athlete      ; rdfs:range :Team.
:coachOf     rdfs:domain :Coach        ; rdfs:range :Team;
             rdfs:subPropertyOf :memberOf.

:competitor  rdfs:domain :Event        ; rdfs:range :Team.
:location    rdfs:domain :Event        ; rdfs:range :Place.

:e1 :competitor :t1 , t2 .
:e2 :competitor :t2 , t3 .

:p1 :coachOf :t2 .      :p5 :coachOf :t1 .

:p1 :memberOf :t1 , t3 . :p2 :memberOf :t1 . :p3 :memberOf :t2 .
:p4 :memberOf :t2 .      :p5 :memberOf :t3 . :p6 :memberOf :t4 .

```

Question 4 (6 points)

Exprimez les requêtes suivantes en SPARQL sur *sports.ttl*. On suppose que le graphe est saturé avant

5. Les athlètes qui participent à tous les évènements.

6. Les équipes qui ne se sont jamais rencontrées.

Solution: Toutes les équipes participant à l'évènement :e1

```
select distinct ?x
from <sportsinf.ttl>
where { :e1 :competitor ?x . }
```

Toutes les équipes qui participent à l'évènement :e1 mais pas à l'évènement :e2.

```
select distinct ?x
from <sportsinf.ttl>
where { { :e1 :competitor ?x } minus { :e2 :competitor ?x } . }
```

Tous les athlètes qui participent à l'évènement :e1.

```
select distinct ?s
from <sportsinf.ttl>
where { :e1 :competitor ?x .
       ?s :memberOf ?x . }
```

Tous les athlètes qui participent à l'évènement :e1 mais pas à l'évènement :e2.

```
select distinct ?s
from <sportsinf.ttl>
where { { :e1 :competitor ?x . ?s :memberOf ?x }
       minus { :e2 :competitor ?y . ?s :memberOf ?y } . }
```

Solution: Tous les athlètes qui participent à tous les évènements.

```
select distinct ?s
from <sportsinf.ttl>
where { { ?s a :Athlete }
        minus { { ?e a :SportsEvent . ?s a :Athlete }
                minus { ?e :competitor ?t . ?s :memberOf ?t } } }
```

3 Plans de Jointure et Optimisation (4 pts)

On veut exécuter la requête SPARQL suivante:

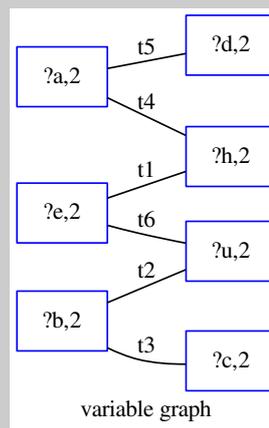
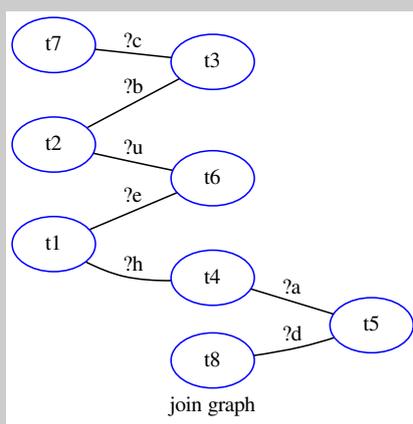
```
select distinct ?e ?a ?b ?h ?u
from <sportsinf.ttl>
where {
t1      ?e :homeTeam ?h .
t2      ?b :memberOf ?u .
t3      ?b :memberOf ?c .
t4      ?a :memberOf ?h .
t5      ?a :memberOf ?d .
t6      ?e :awayTeam ?u .
t7      :e1 :competitor ?c .
t8      :e1 :competitor ?d .
}
```

Question 5 (2 points)

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 (t_1, t_2, \dots, t_8).

Réponse : Graphe de jointure et graphe de variables.

Solution:



Question 6 (1 point)

Expliquez pourquoi l'évaluation de la requête prend beaucoup de temps si on applique les filtres dans l'ordre d'apparition dans la clause WHERE, c.a.d. en appliquant le plan linéaire ((((((t1, t2), t3), t4), t5), t6), t7), t8).

Réponse :

Solution: Le plan de jointure calcule plusieurs produits cartésiens et évalue les motifs de triplets (t7, t8) sélectifs à la fin.

Question 7 (1 point)

Donnez un meilleur plan linéaire que le précédent et expliquez pourquoi il est meilleur.

Réponse :

Solution: On doit commencer avec des triplets de faible cardinalité (t7, t8) et générer un plan qui évite les produits cartésiens : ((((((t8,t5),t4),t1),t6),t2),t3),t7)

4 Requêtes et Ontologies (5 pts)

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

$Athlete \equiv Person \sqcap \exists memberOf.Team$ (1)

$Coach \equiv Person \sqcap \exists coachOf.Team$ (2)

$member \equiv memberOf^-$ (3)

Question 8 (1 point)

Décrivez les concepts *Athlete* et *Coach* et le rôle *member* en français (ou en anglais).

Réponse :

Question 9 (2 points)

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

Réponse :

Solution:

Question 10 (2 points)

Traduisez la requête SPARQL suivante en une nouvelle requête SPARQL qui calcule la même réponse *sans* sans utiliser les concepts *Athlete* et *Coach* et le rôle *member*.

```
select ?s
where { ?s :member ?c .
        ?c rdf:type :Coach . }
```

Réponse :

Solution: `select ?a where ?c :memberOf ?s . ?c rdf:type :Person; :coachOf ?s2 .`