

Cet examen comprend 2 questions sur un total de 20 points. Les points sont à titre indicatif.

Question:	1	2	Total
Points:	15	5	20
Score:			

1. Question[15] Soit la base de connaissances  $KB$  définie par :

1. Etudiant  $\sqsubseteq \exists$  assiste.Cours
  2. Cours  $\sqsubseteq \exists$  assistePar (Enthousiaste  $\sqcap$  Etudiant)
  3. Etudiant(aStudent)
  4. Cours(aCours)
  5.  $\neg$ Enthousiaste(aStudent)
- (a) (2 points) Donnez la définition de  $KB$  en OWL, vous pouvez utiliser la syntaxe de Turtle ou RDF/XML.

**Solution:**

```
# Object Properties
:assiste rdf:type owl:ObjectProperty ;
    rdfs:range :Cours ;
    rdfs:domain :Etudiant .

:assistePar rdf:type owl:ObjectProperty ;
    owl:inverseOf :assiste .

# Classes
:Cours rdf:type owl:Class ;
    rdfs:subClassOf [ rdf:type owl:Restriction ;
        owl:onProperty :assistePar ;
        owl:someValuesFrom [ rdf:type owl:Class ;
            owl:unionOf( :Enthousiaste
                :Etudiant
            )
        ]
    ] ;
    owl:disjointWith :Etudiant .

:Enthousiaste rdf:type owl:Class .
:Etudiant rdf:type owl:Class ;
    rdfs:subClassOf [ rdf:type owl:Restriction ;
        owl:onProperty :assiste ;
        owl:someValuesFrom :Cours
    ] .

:normalEtudiant rdf:type owl:Class ;
    rdfs:subClassOf [ rdf:type owl:Restriction ;
        owl:onProperty :assiste ;
        owl:onClass :Cours ;
        owl:minQualifiedCardinality
            "3"^^xsd:nonNegativeInteger
    ] .
```

```

:serieuxEtudiant rdf:type owl:Class ;
                  rdfs:subClassOf [ rdf:type owl:Restriction ;
                                    owl:onProperty :assiste ;
                                    owl:allValuesFrom :Cours
                                    ] .
#      Individuals
:aStudent rdf:type owl:NamedIndividual ,
             [ rdf:type owl:Class ;
               owl:complementOf :Enthousiaste
             ] .

:aSudent rdf:type :Etudiant ,
              owl:NamedIndividual .

[ rdf:type [ rdf:type owl:Class ;
             owl:complementOf :Enthousiaste
           ]
] .

```

(b) On souhaite ajouter à *KB* les définitions ci-dessous. Donnez la description en DL de ces définitions.

(a) (1 point) Le concept : *serieuxEtudiant* rassemblant les étudiants qui assistent à tous les cours.

**Solution:**  $\text{serieuxEtudiant} \sqsubseteq \text{Etudiant} \sqcap \forall \text{ assiste Cours}$

(b) (1 point) L'individu *aStudent* est un *serieuxEtudiant*.

**Solution:**  $\text{aSudent} : \text{serieuxEtudiant}$ .

(c) (1 point) Le concept : *normalEtudiant* rassemblant les étudiants qui assistent au moins à 3 trois cours.

**Solution:**  $\text{normalEtudiant} \sqsubseteq \geq 3 \text{ assiste Cours}$

(d) (1 point) Les concepts *Etudiant* et *Cours* sont disjoints.

**Solution:**  $\text{Cours} \sqsubseteq \neg \text{Etudiant}$

(e) (1 point) Le domaine et le range du rôle *assiste*.

**Solution:**  
 $\exists \text{ assiste Thing} \sqsubseteq \text{Etudiant}$   
 $\top \sqsubseteq \forall \text{ assiste Cours}$

(f) (2 points) Est-ce que le concept *serieuxEtudiant* est équivalent au concept :  
 $\text{Etudiant} \sqcap \exists \text{ assiste.Cours}$ . Pourquoi ?

**Solution:**  
Non, car *serieuxEtudiant* assiste à tous les cours et que des cours tandis que le concept dans cette question il suffit d'assister à un seul cours.

(g) (1 point) Est-ce que c'est nécessaire de définir le range du rôle *assiste*. Pourquoi ?

**Solution:** oui, car la base de connaissance ne permet pas de déduire le range de *assiste*

(h) (2 points) Ecrivez la requête SPARQL *Q1* qui donne les étudiants et les cours auxquels ils ont assisté. Donnez les résultats de l'exécution de *Q1* sur *KB* sans inférence et avec inférence, justifiez votre réponse

**Solution:**  

```

select ?et ?c
where {
  ?et rdf:type :Etudiant .

```

```
?c rdf:type :Cours ?  
?et :assiste ?cours .  
}
```

- (i) (2 points) On définit  $KB' = \{KB, Etudiant \sqsubseteq Personne, Personne(aPersonne)\}$ . Donnez la requête SPARQL Q2 qui donne toutes les personnes. Donnez les résultats de l'exécution de Q2 sur  $KB'$  sans inférence et avec inférence, justifiez votre réponse.

**Solution:**

```
select ?p  
where {  
  ?p rdf:type :Personne  
}
```

Sans inférence : rien. Avec inférence : aPersonne et aStudent, car Etudiant est subClass de Personne

- (j) (1 point) Est-il possible de modéliser  $KB$  en RDFS ? Justifiez votre réponse.

**Solution:** Non, intersection, négation,...

## 2. Question [5]

- (a) (1 point) Citez deux avantages liées à la sémantisation de données ouvertes.

**Solution:** Permet l'intégration facile de données, partage de données, faire de requêtes sur les données..

- (b) (2 points) Donnez les définitions informelles de : SPARQL endpoint, une requête fédérée, DBpedia.

**Solution:** *SPARQL endpoint* : SPARQL endpoint is a URI to which you send a SPARQL request (Query, Update etc) and to which the endpoint responds appropriately. the endpoint implements the "P" in SPARQL - the protocol as defined in <http://www.w3.org/TR/rdf-sparql-protocol/>. C'est un web service qui implémente le protocole SPARQL. Il reçoit des requêtes SPARQL et envoie les résultats. *une requête fédérée* : requête SPARQL qui accède à plusieurs SPARQL endpoints.

*DBpedia* Une base de connaissances générales construites à partir de Wikipedia

- (c) (2 points) OWL DL est un langage pour modéliser les ontologies, donnez deux différences entre OWL DL et UML.

**Solution:** OWL DL a une définition formelle et permet l'inférence. Elle permet de modéliser l'équivalent de diagramme de classe UML plus général que OWL DL, il permet de définir use-case, diagramme d'activité ..