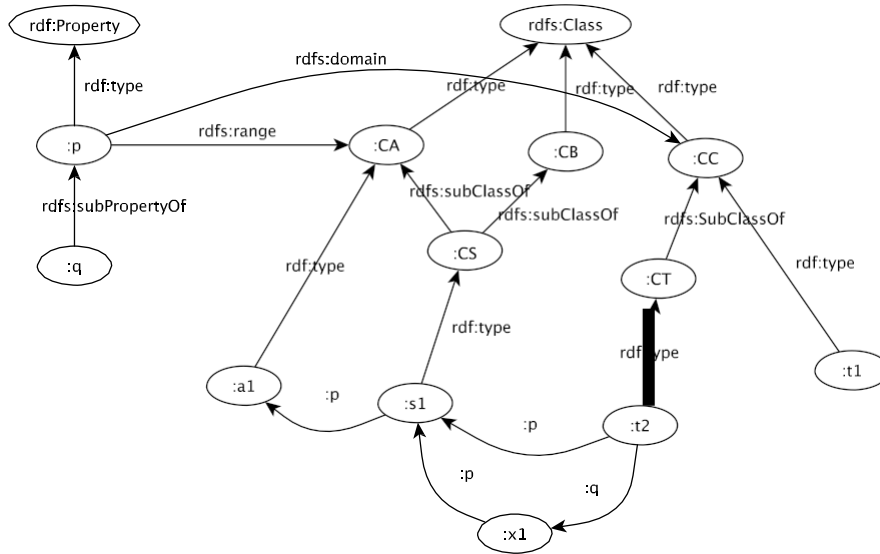


WRITTEN EXAM: SOME SAMPLE QUESTIONS

Q: SPARQL

Consider the following graph RDF/S // On considère le graphe RDF suivant



1. What will be the result of executing the following queries on this graph

- without RDFS entailment
- with RDFS entailment

Quel sera le résultat de l'exécution des requêtes suivantes sur ce graphe

- sans inférence RDFS
- avec inférence RDFS

- select ?s where { ?s :p ?o }
- select ?w where { ?w a :CA }
- select ?x where { ?y :p ?x. filter not exists { ?y a :CC } }
- select ?x where { ?x :p ?y. ?y a :CA }

2. Write SPARQL queries that correspond to the following questions:

- find all the classes *C* that have at least one member that is connected through `:p` to a member of a subclass of *C*
trouver toutes les classes C dont au moins un membre est connecté par la propriété :p à un membre d'une sous-classe de C
- find all the members of the class `:CC` that are connected to at most one node through property `:p`.
trouver tous les membres de la classe :CC qui sont connecté à au plus un noeud par la propriété :p.

Q: SPARQL rewriting

A SPARQL endpoint S has an RDF schema that defines the classes $s:Person$ and $s:Farmer$ and the property $s:hasAncestor$.

For this endpoint a query to find all the ancestors of a person that are/were farmers can be expressed as:

```
Q: select ?a
    where { ?a a s:Person. ?p a s:Person.
            ?p s:hasAncestor ?a. ?a a s:Farmer }
```

In another endpoint T , the schema has the classes: $t:LivingPerson$, $t:DeadPerson$, $t:Cultivator$, and the properties $t:hasFather$ and $t:hasMother$.

Rewrite Q in order to obtain an (almost) equivalent query for the endpoint T .

Q: DL modeling

An OWL ontology contains the following class hierarchy, properties and individuals:

Classe hierarchy:

```
graph TD
    Place --> Castle
    Place --> HauntedCastle
    Place --> BedAndBreakfast
    Place --> GuestHouse
    Place --> PerchedHut
    Entity --> Ghost
    Entity --> Tree
    Purpose --> Providing
    Object --> Accomodation
    Object --> Breakfast
    Country --> Scotland
```

Properties:

locatedIn
frequentedBy
hasPurpose
hasObject

Individuals:

Scotland (instance of Country)

Hint: Here is the description of a Market with a similar vocabulary:



Write axioms (in DL or Manchester syntax) to express the following elements of domain knowledge:

1. A haunted castle is a castle frequented by ghosts
Un château hanté est un château fréquenté par des fantômes
2. Every castle located in Scotland is frequented by at least 2 ghosts
Tout château situé en Ecosse est fréquenté par au moins 2 fantômes

3. A bed and breakfast is a place whose purpose is providing accomodation and breakfast and which is located in a guest house

Un bed and breakfast est un endroit qui a pour but de fournir hébergement et petit déjeuner et qui est situé dans une maison d'hôtes

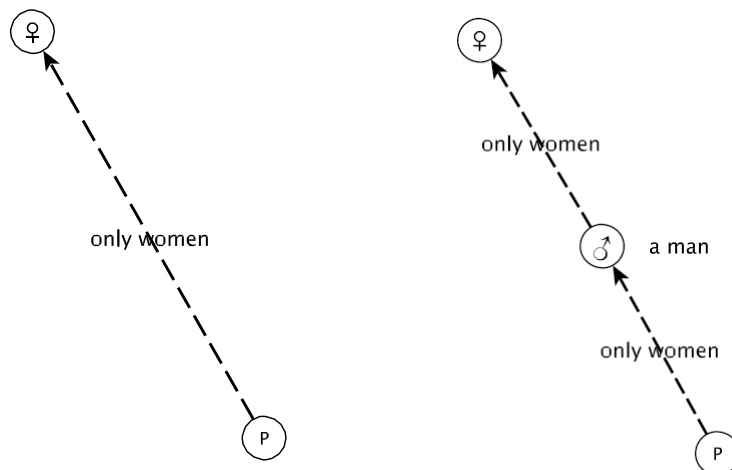
4. A perched hut is a place located in a tree whose purpose is providing accomodation

Une cabane perchée est un endroit situé dans un arbre qui a pour but de fournir un hébergement

Q: SWRL modeling

Define a vocabulary (classes and properties) and SWRL rules for representing the following definitions:

1. The uncle of a person is a brother of the mother or a brother of the father of this person
L'oncle d'une personne est un frère de la mère ou un frère du père de cette personne
2. The women-only-ancestors of a person are his/her mother, mother of the mother, mother of the mother of the mother, etc. (see figure, left)
Les ancêtres-femmes-seulement d'une personne sont sa mère, la mère de sa mère, la mère de la mère de sa mère, etc. (à gauche sur la figure)
3. The almost-women-only-ancestors of a person are his/her ancestors connected to him/her through a chain containing only women except for one man (see figure right)
Les ancêtres presque-seulement-femmes d'une personne sont ses ancêtres femmes liées à elle par une chaîne ne comprenant que des femmes, à l'exception d'un homme (à droite sur la figure).



Q: SWRL to DL

An ontology contains the following SWRL rules:

Une ontologie contient les règles SWRL suivantes:

```
p(?X, ?Y) -> q(?Y, ?X)
q(?X, ?Y) -> p(?Y, ?X)
```

```
Car(?X), hasMaker(?X, ?M), inCountry(?M, Italy) -> ItalianCar(?X) .
```

```
Person(?X), hasChild(?X, ?Y), Person(?Y), hasChild(?X, ?Z),
                                                                    Person(?Z)
, differentFrom(?Y, ?Z) -> PW2C(?X) .
```

Your goal is to replace these rules with equivalent OWL axioms (that produce the same inferences). These axioms can be of the form <expression> subClassOf <expression>, <expression> subPropertyOf <expression>, <property> isInverseOf <property>, <property> isTransitive, etc.

Votre but est de remplacer ces règles par des axiomes OWL équivalents (qui produisent les mêmes inférences). Ces axiomes peuvent être de la forme <expression> subClassOf <expression>, <expression> subPropertyOf <expression>, <property> isInverseOf <property>, <property> isTransitive, etc.

Example

```
Car(?X), driver(?X, ?D) -> Person(?D) .
```

can be replaced by // *peut être remplacé par*

```
Car subClassOf driver only Person
```