SEMANTIC DATA MANAGEMENT EXAM

13th of June 2024. *The exam will take **2 hours**. Answer each question in the provided space.* <u>Answers out of such space will not be considered</u>. Further, clearly read the instructions how to answer. Answers not following the required format might not be considered.

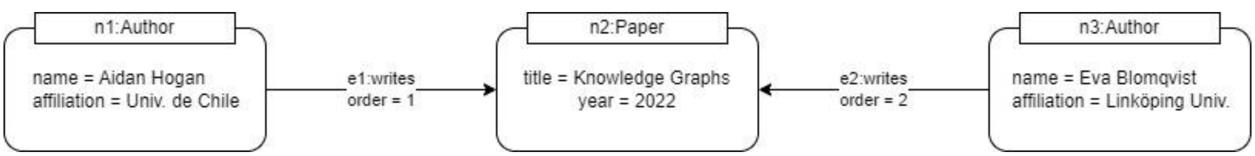
Name: ……….

QUESTION 1. GRAPH OPERATIONS [2p]

a) Given the following **navigational graph pattern**:

Draw a *minimal* (i.e., minimize the number of nodes and edges not present in the **match**) property graph that would return at least one **match**. Draw your answer in the space below.

b) Given the following property graph:



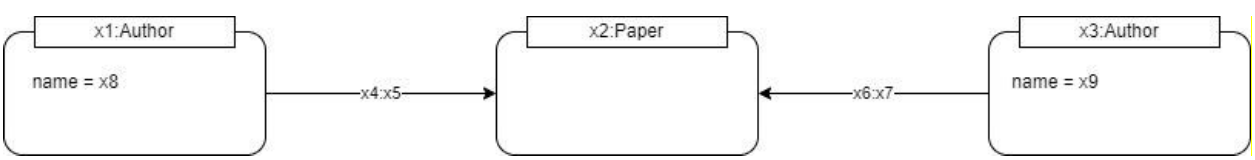
Propose a **bgp (basic graph pattern)** that would return different matches when considering isomorphism-based and homomorphism-based semantics:

SOLUTION

a)

(Person: Person1) <- [:Loves]- (Person1)   (instead of “Loves” it could be “Likes”)

b)



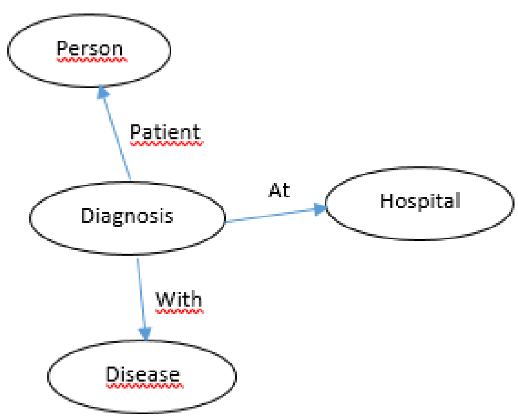
With homomorphism, a valid match would be one in which x1=x3. This cannot happen in isomorphism.

QUESTION 2. GRAPH MODELING [2p]

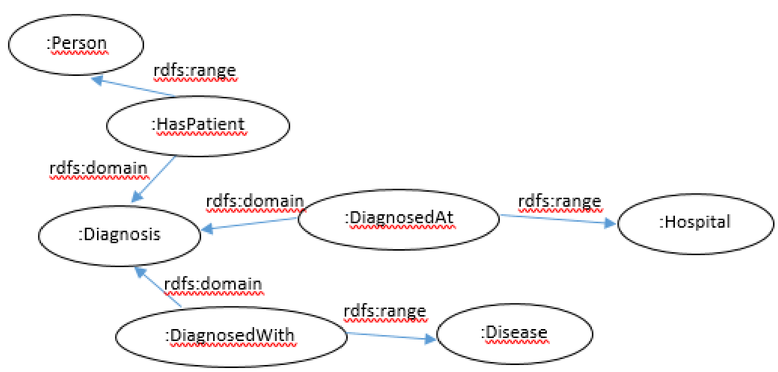
A company working with health data wants to model some of its data using graphs. The piece of data you are asked to model is the following:
"A person is diagnosed with a certain disease at a specific time (diagnosis time) and at a specific hospital."

Provide a data model for property graphs and another one for knowledge graphs (in RDFS). Follow the good practices discussed in the course when modeling in one or another model and use the allocated space for each model.

Property Graph



Knowledge Graph



Is there any substantial difference between both solutions? Justify your answer.

SOLUTION

There is no big difference in this case. Even if in property graphs we can benefit from edge attributes, hospital and disease are not a good candidate to be an edge attribute, since it is a relevant concept that is better model as a nodel (i.e., class). Diagnosis time can be kept as an edge attribute (in that case, between diagnosis and hospital). But in any case, this statement requires reification in both models.

QUESTION 3. KNOWLEDGE GRAPHS: RDFS [2p]

a. Two teammates, participating in data modeling tasks at their company, Amateus, which analysis data related to tourism, are conflicted because each of them is proposing a different RDFS TBOX for the graph to be created.

Person#1 proposes the following TBOX triples:

```
myC:person myC:travelsTo myC:country
```

Person#2 proposes the following TBOX triples:

```
myC:travelsTo rdfs:domain myC:person
myC:travelsTo rdfs:range myC:country
```

Is there any difference between the two proposals? Justify your answer (*yes/no answers without a proper justification will not be considered*).

b. The same two colleagues are conflicted again when modeling *destinations*. A destination, from the tourism point of view, it is not only about countries but also touristic regions or places without a clear administrative entity (which we call landmarks). For example, the Cappadocia or the Andes would be touristic regions, while Taj Mahal would be a touristic place. For this reason, Person#2 proposes the following classes and taxonomy:

```
myC:country rdf:type rdfs:Class
myC:touristicLandmark rdf:type rdfs:Class
myC:touristicRegion rdf:type rdfs:Class
myC:touristicPlace rdf:type rdfs:Class
myC:touristicRegion rdfs:subClassOf myC:touristicLandmark
myC:touristicPlace rdfs:subClassOf myC:touristicLandmark
```

Since a destination can be a country or a touristic landmark, she proposes these two additional triples:

```
myC:travelsTo rdfs:range myC:country
myC:travelsTo rdfs:range myC:touristicLandmark
```

#Person#1 sighs and bemoans for not paying more attention during the Semantic Data Management course she took some years ago. Could you help them and provide a *better* solution? Here, better means that you can save writing some triples and also modeling the same information with precise semantics. Provide the required information to interpret your solution if you make any assumption.

SOLUTION

A)

Using predefined rdfs properties such as domain and range generates inference out of these triples (rdf:type property, rdf:type class and also domain-specific inference when the ABOX triples are inserted) if the RDFS regime entailment is activated.

B)

```
myC:touristicRegion rdfs:subClassOf myC:touristicLandmark
myC:touristicPlace rdfs:subClassOf myC:touristicLandmark
myC:touristicLandmark rdfs:subClassOf myC:destination
myC:country rdfs:subClassOf myC:destination
myC:travelsTo rdfs:range myC:destination
```

Alternatively, you can add more detail as follow:

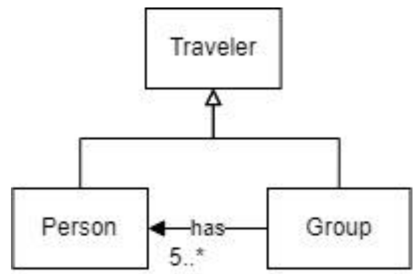
```
myC:travelsToTP rdfs:range myC:touristicPlace
myC:travelsToTR rdfs:range myC:touristicRegion
myC:travelsToC rdfs:range myC:country
myC:travelsToTP rdfs:subPropertyOf myC:travelsTo
myC:travelsToTR rdfs:subPropertyOf myC:travelsTo
myC:travelsToC rdfs:subPropertyOf myC:travelsTo
```

We can save all the rdf:type triples by activating inference. The complete taxonomy helps constraining the data most appropriately.

Further, the two-range definition for travelsTo is wrong, since for any ABOX triple whose property is travelsTo its subject will be asserted as country AND touristicLandmark (which does not make sense).

QUESTION 4. DESCRIPTION LOGICS AND OWL [2p]

The two colleagues from the previous question gave up on RDFS and move on to Description Logics and OWL. They came up with the following conceptual schema for the traveler taxonomy:



This means a traveler is either an individual person or a group. They do not consider any other kind of traveler. Finally, they define a group to be of minimum size 5 (i.e., a group has 5 or more people).

a) Model in Description Logics **as many constraints as possible** from the conceptual schema above:
b) Now, express the DL axioms above in OWL.

SOLUTION DL

Person $\sqsubseteq$ Traveler

Group $\sqsubseteq$ Traveler

Person $\sqsubseteq$ $\neg$Group (*disjointness*)

Traveler $\sqsubseteq$ Person $\sqcup$ Group (*completeness*)

Group $\sqsubseteq$ >=5 has (*cardinality*)

$\exists$has $\sqsubseteq$ Group (*domain*)

$\exists$has^{-} $\sqsubseteq$ Person (*range*)

SOLUTION OWL

myC:person rdfs:subClassOf myC:traveler

myC:group rdfs:subClassOf myC:traveler

myC:has rdfs:domain myC:group

myC:has rdfs:range myC:person

_:a owl:complementOf myC:group

myC:person rdfs:subClassOf _:a

_:b owl:unionOf (myC:person, myC:group)

myC:traveler rdfs:subClassOf _:b

_:c rdfs:subClassOf owl:restriction

_:c owl:onProperty myC:has

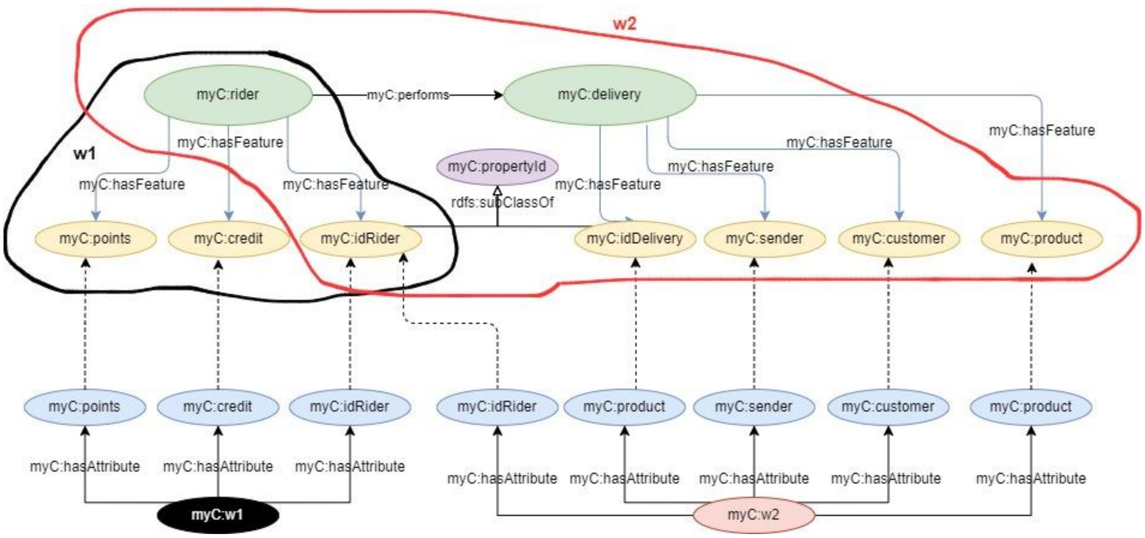
_:c owl:minCardinality 5

myC:group rdfs:subClassOf_:c

QUESTION 5. DATA INTEGRATION [2p]

A delivery company, Vlogo, developed a **graph-based virtual data integration system** to integrate the data sent by all the monitoring devices embedded in the riders's apps. In this setting, a sender sends a product to a customer in what we call a delivery. A delivery is performed by a rider. A rider has a credit (while there is credit, they can take deliveries) and points assigned by the customers (kind of feedback). For a delivery, we track the rider, the sender, the customer (i.e., receiver) and the product sent.

In the figure below, you can see the system created: global graph, source graphs and the LAV mappings between them (the dashed lines are *owl:sameAs* relationships).



Now, it is time to use the system! The first query they want to pose is the following: "*the rider points p for those who ever delivered from sender s*". Basically, they want to study if certain senders guarantee higher points (regardless of the customer). For this matter, they wrote the following query:

```
SELECT ?p,?sd
WHERE {
    ?p rdf:type myC:points
    ?r myC:hasFeature ?points
    ?r rdf:type myC:rider
    ?r myC:performs ?d
    ?d rdf:type myC:delivery
    ?d myC:hasFeature ?sd
    ?sd rdf:type myC:sender
}
```

Execute the rewriting algorithm and generate the algebraic expression of the operations to be executed on the wrappers.

SOLUTION:

$$\Pi_{points,\ sender}\ (\rho_{W1.points\text{->}p}$$

$$\rho_{W2.sender\text{->}sd}$$

$$(\sigma_{W1.idRider=W2.idRider}(W1 \times W2)))$$