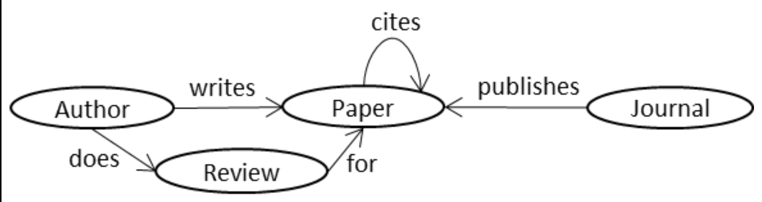
SEMANTIC DATA MANAGEMENT EXAM

12th of June 2023. *The exam will take **2 hours**. Answer each question in the provided space.* Answers out of such space will not be considered. Further, clearly read the instructions how to answer. Answers not following the format set might not be considered.

Name: ………

QUESTION 1. PROPERTY GRAPHS [4p]

We want to create a property graph modeling scientific publication in journals. More precisely, authors write papers that are published in journals. A journal publishes papers in terms of volumes. There can be various volumes of a journal per year. A paper can be written by many authors, and one of them acts as corresponding author. We also want to store the organization each author is affiliated to. A paper can be cited by another paper. Finally, we also want to include in the graph the concept of review. Each paper has a set of reviewers, who are also authors that have published other papers. A possible solution could be the following (note that we are using a simplified notation that shows the types of nodes and edges instead of the instances, and properties are listed separately):



Properties:
Author: name, is_corresponding, affiliation
Paper: DOI, title, year
Journal: name
Review: text, score
publishes: volume, year

This solution is not fully correct, and it is not the best in terms of quality (non-redundancy, maintainability, understandability, efficiency of possible queries, …).

a. **Provide a graph** that solves the issues in the previous solution. **Justify** your changes.

b. For each of the following Cypher queries (and assuming the initial graph was correct), **choose** the **theoretically optimal graph operation** (see box below) that can be applied to solve it. **Justify** your answers.

Adjacency: $\text{Adjacency}(n) = \bar{N}$
$n_i \in \bar{N} \iff \exists e_1 \mid \rho(e_1) = (n_i, n) \vee \rho(e_1) = (n, n_i)$

Reachability: $\text{Reachability}(n_{or}, n_{dest})$ is **true** $\iff \exists \text{Walk}(n_{or}, n_{dest})$
$\text{Walk}(n_{or}, n_{dest}) = (e_1 \ldots e_m) \mid \exists n_1 \ldots n_{m-1}, \rho(e1) = (n_{or}, n_1), \rho(e2) = (n_1, n_2)$
$\ldots \rho(e_m) = (n_{m-1}, n_{dest})$

Label-constrained reachability: $x \xrightarrow{\alpha} y$
where x, y can be variables, nodes, or a mix, and α is a regular expression over *Lab*

Pattern Matching: Based on Basic Graph Patterns (bgps). A bgp is a property graph where variables can appear in place of any constant.

1. MATCH (a: Author) -[:writes]-> (Paper) <-[:publishes]- (j: Journal)
   RETURN a, j
2. MATCH (a: Author) -[:writes]-> (Paper {title: ‘The DL-Lite family’}) <-[:publishes]- (j: Journal)
   RETURN a, j
3. MATCH (p: Paper {title: ‘Linking data to ontologies’}) <-[:cites*]- (p2: Paper)
   RETURN p, p2
4. MATCH (:Journal {name: ‘Future Generation Computer Systems’})--> (p)
   RETURN p

Use the remaining of this page to provide your solutions to question 1.

QUESTION 2. KNOWLEDGE GRAPHS [4p]

a. Consider the following triples in a Knowledge Graph whose regime entailment is RDFS (as we saw it in the lectures):

TBOX (or schema):

```
myURL:A rdfs:subClassOf myURL:B

myURL:A rdfs:subClassOf myURL:C
```

ABOX (or instances):

```
myURL:a rdf:type myURL:A
```

Write ALL the inferred triples generated by the RDFS regime entailment for the above triples:

b. Express in Description Logics the TBOX and ABOX described in the previous section.

TBOX T:

ABOX A:

c. Considering your solution for section b **plus** the following DL statement: $B \sqsubseteq \neg C$

Is the resulting **interpretation** a model of the Ontology $O = < T, A >$? Justify your answer.

d. We want to model the following constraints in a knowledge graph:

*"A food **delivery contains** between **three** and **ten products**"*

Express these cardinality constraints in Description Logics using at most the concepts highlighted in the above constraints and do not create new ones. For the role *contains* assume it relates a delivery to a product (in this order).

e. Express the DL cardinality constraints from section d in OWL:

QUESTION 3. DATA INTEGRATION [2p]

We aim at developing a **graph-based virtual data integration system** in on our company, which solves a logistics problem to small companies or customers by bringing products from A to B in what we call a delivery. Our company is a start-up born as a digital company and therefore we track absolutely any movement: our riders (people performing the delivery), the products contained and the customers participating as senders or receivers of a delivery.

Consider a simplified version of the company data sources. In this exercise, we will focus on two of them: the delivery information and the set of files generated by the riders when performing the delivery between A and B. For the sake of simplicity, consider the following sources (in brackets, the acronym for each of the relevant attributes; { } denotes a set):

- A PostgreSQL database storing the delivery information. It contains the following information: a sender id (S), a receiver id (R), a list of products sent ({P}) and the delivery id (ID).
- A JSON file per delivery. For this exercise, you can consider a single file with the following information: the rider id (RID), the delivery id (DID), the pick-up time (PT), the delivery time (DT) and a list of speeds ({SP}) (every 30” the application monitors the speed and stores it in the list).

The figure below shows the integrated graph created (which you should not modify). First, we need to create the wrappers exposing the source data to the system.

a. Define the wrappers. Realise the description above is that of the sources. Here, you must create the required wrappers to align them with the integrated graph (see next page). You can use the usual signature for these exercises (e.g., W1 (A, B, C, D)). Use the acronyms introduced above to refer to the attribute from the sources. If you define any new attribute/acronym be sure to properly define it in the next question.

$W_{delivery}$ (

W_{riders} (

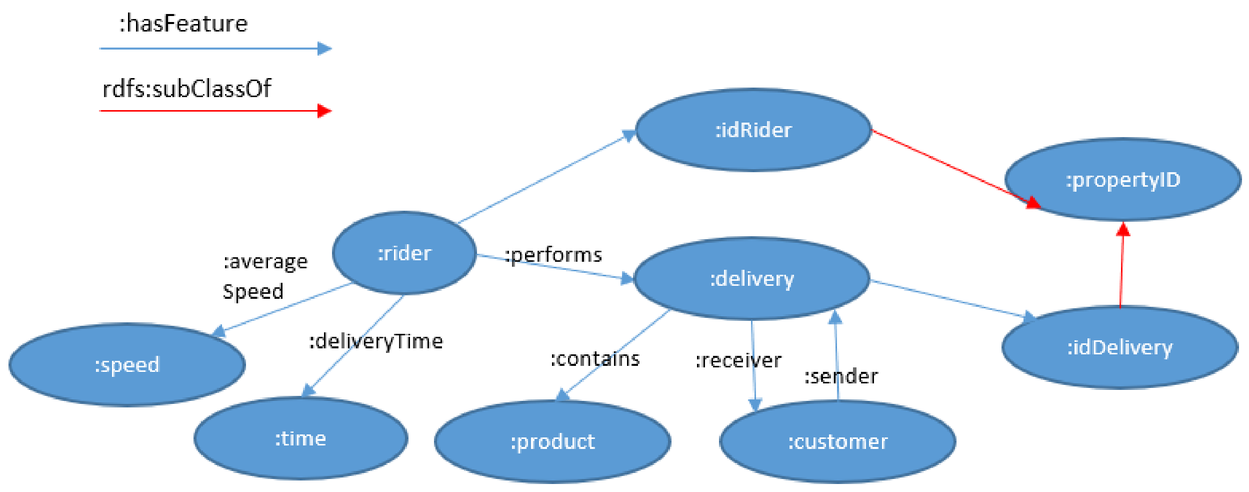
b. Did you create any new attribute/acronym in the wrappers signature? If so, for each of them, explain how it would be computed from the attributes/acronyms described in the sources:

In the next page, you can see the global (integration) schema generated in the company. All the properties displayed in the global schema (e.g., *:performs*) are defined as <*property*> `rdfs:subPropertyOf :hasFeature` and have been defined this way to facilitate their understanding.

c. Create the source graphs corresponding to the two wrappers identified. Follow the same notation and format as in the lecturers and draw them **below** the *source graphs* line.

d. Draw the **LAV mappings** between the global and local levels required to make the data integration system work. **Draw the mappings following the notation used in the lectures (i.e., a named graph and its corresponding *owl:sameAs* triples)**. If you can use two different colours for each wrapper, please, do so.

Global graph ____



____**Source graphs**