Name: .....

Task 1 (5 Points)

Let the prefix `xsd` be defined as `http://www.w3.org/2001/XMLSchema#`.

a) Which of the following three RDF literals describe the same value?

   (a) `"2"^^xsd:integer`

   (b) `"2"^^xsd:float`

   (c) `"2.0"^^xsd:decimal`

b) Which of the following three RDF literals describe the same value?

   (a) `"2021-03-16T09:20:00"^^xsd:dateTime`

   (b) `"2021-03-16T09:20:00+01:00"^^xsd:dateTime`

   (c) `"2021-03-16T08:20:00Z"^^xsd:dateTime`

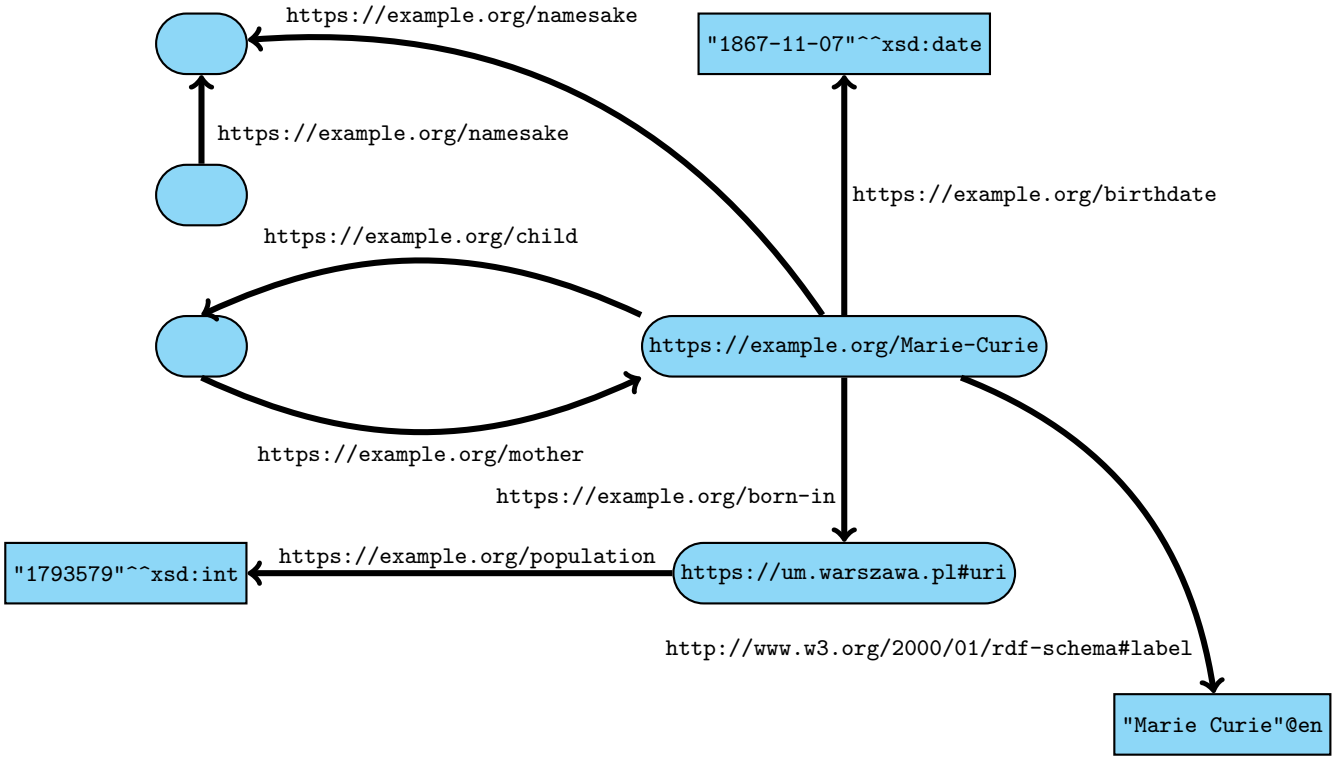
c) Define the term “RDF graph”. You may use the terms “IRI”, “blank node”, “literal”, and “Unicode” in your definition.

Name: .....

Name: .....

Task 2 (6 Points)

Use the Turtle format to encode the following RDF graph, using the base `https://example.org/` and the `xsd` (`http://www.w3.org/2001/XMLSchema#`) `rdfs` (`http://www.w3.org/2000/01/rdf-schema#`) prefixes. Take advantage of syntactic abbreviations wherever possible.

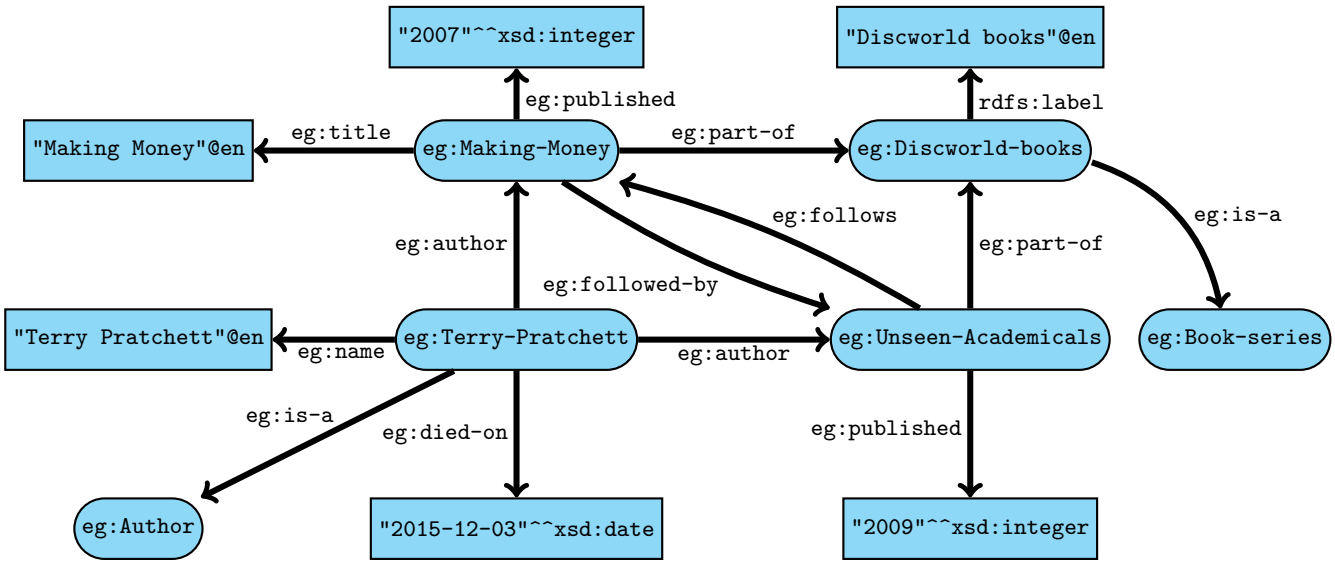


Name: .....

Name: .....

Task 3 (12 Points)

Consider an RDF graph describing authors and book series, using a schema as described by the following image (i.e., the image shows just a tiny portion of the whole graph, which contains information on many more book series, authors, and books):



Write SPARQL queries that find

a) all books belonging to the Discworld series;

b) all authors of book series, ordered by the maximal number of books they contributed to any series (note that multiple authors may contribute books to a series, and that authors may contribute books to multiple series); and

c) all book series consisting only of books whose authors have died. Note that multiple authors may contribute books to a series

Name: .....

Name: .....

Task 4 (5 Points)

a) Give a formal definition of a *lean RDF graph*. Briefly describe the intuitive significance of this notion.

b) Which of the following RDF graphs is not lean? Specify an instance of it that witnesses its non-leanness.

(i)
```
eg:s   eg:p   eg:o .
_:1    eg:p   _:1  .
```

(ii)
```
eg:s   eg:p   _:2  .
_:1    eg:p   eg:o .
```

(iii)
```
eg:s   eg:p   eg:s .
_:1    eg:p   [ eg:p   [] ] .
```

(iv)
```
eg:s   eg:p   eg:o .
_:1    eg:p   [ eg:p   [] ] .
```

Name: .....

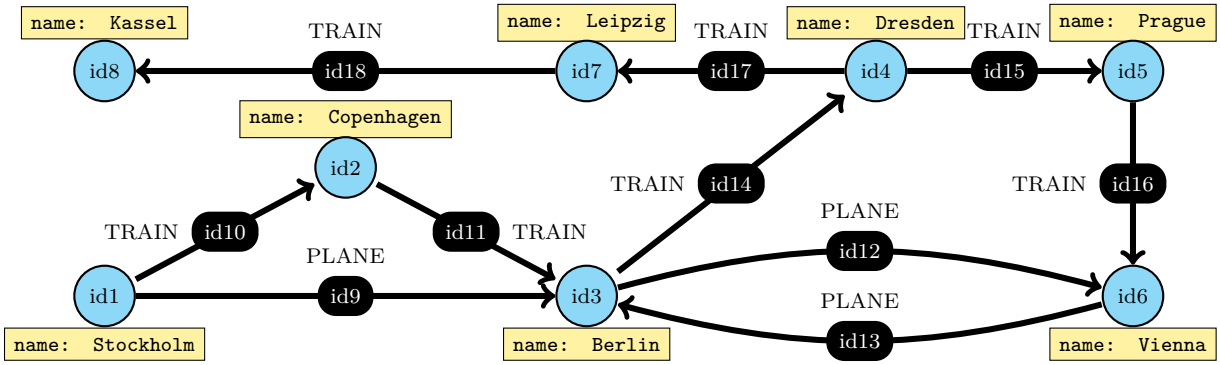
Name: .....

Task 5 (8 Points)

a) Consider a property graph that uses the `HAS_CHILD` relationship type to model parent–child relationships. Write a Cypher query that finds persons and their great-grandparents.

b) Which answers does the following Cypher query produce on this graph?

```
MATCH p = (s {name: "Stockholm"})-[:TRAIN*]-(a)-[:PLANE*1..]->
          (b)-[:TRAIN*0..]-(d {name: "Vienna"})
RETURN [ n IN nodes(p) | n.name ]
UNION ALL MATCH p = (s {name: "Stockholm"})-[:TRAIN*]-(d {name: "Vienna"})
RETURN [ n IN nodes(p) | n.name ]
```



Name: .....

Name: .....

Task 6 (11 Points)

a) Consider a schema consisting of two unary predicates `first` and `last`, and of a binary predicate `next`. Let D be a database of that schema, encoding a linear order of the form $1 < 2 < 3 < \cdots < n-1 < n$ using facts

$$\texttt{first}(1) \qquad \texttt{next}(1,2) \qquad \texttt{next}(2,3) \qquad \cdots \qquad \texttt{next}(n-1,n) \qquad \texttt{last}(n)$$

Write a Datalog program P such that $\langle P, \texttt{Result}\rangle$ derives `Result` over D iff the linear order encoded by D has even length. If D is not of the form described above, the behaviour of P is unspecified.

b) Give a stratification of the following Datalog program P.

```
mother(?x, ?y)        :- triple(?x, wdt:P25, ?y) .
father(?x, ?y)        :- triple(?x, wdt:P22, ?y) .
notSameMother(?x, ?y) :- mother(?x, ?z), ~mother(?y, ?z) .
sameFather(?x, ?y)    :- father(?x, ?z), father(?y, ?z) .
notSameFather(?x, ?y) :- ~sameFather(?x, ?y) .
halfSiblings(?x, ?y)  :- sameMother(?x, ?y), notSameFather(?x, ?y) .
halfSiblings(?x, ?y)  :- sameFather(?x, ?y), notSameMother(?x, ?y) .
```

Name: .....

Name: .....

Task 7 (4 Points)

Sort the following problems by their computational complexity, from the easiest to the hardest. That is, every problem should provably be at most as hard as all the problems following it.

(1) *given* a SPARQL query q, *decide* whether q has a match on the empty RDF graph

(2) *given* a database instance $\mathcal{I}$, *decide* whether a fixed Datalog query $\langle P, \texttt{Result}\rangle$ derives `Result` on $\mathcal{I}$

(3) *given* a simple graph G, *decide* whether G has a 3-colouring

(4) *given* a database instance $\mathcal{I}$ and a Datalog query $\langle P, \texttt{Result}\rangle$, *decide* whether $\langle P, \texttt{Result}\rangle$ derives `Result` on $\mathcal{I}$

Name: .....

Name: .....

Task 8 (5 Points)

Consider the following ShEx schema:

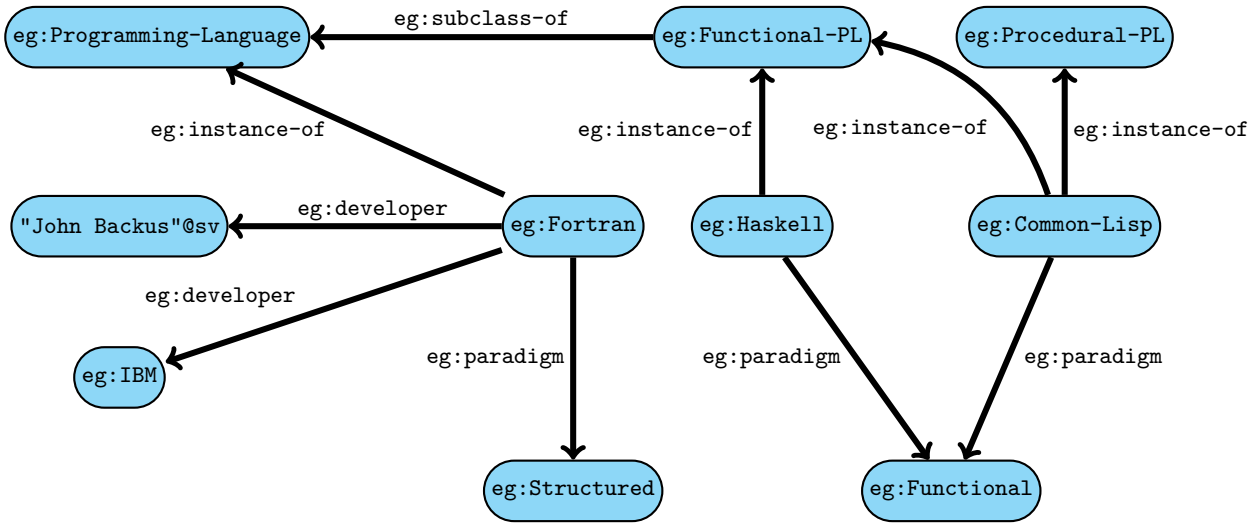
```
eg:programming-language {
  (eg:instance-of @<#programming_language>
  |eg:instance-of @<#subclass_of_programming_language>)+;
  eg:developer IRI*;
  eg:paradigm @<#programming_paradigm>*;
  eg:publication-date LITERAL*;
}

<#subclass_of_programming_language> {
    (eg:subclass-of @<#programming_language> ; eg:subclass-of IRI *) |
    (eg:subclass-of @<#subclass_of_programming_language>; eg:subclass-of IRI *)
}

<#programming_language> [
  eg:Programming-Language
]

<#programming_paradigm> [
  eg:Functional
  eg:Structured
]
```

Validate the RDF graph below according to this schema, using the query map `{ FOCUS eg:paradigm _ } : @eg:programming-language`, i.e., apply the `eg:programming-language` shape to the nodes `eg:Fortran`, `eg:Common-Lisp`, and `eg:Haskell`. Which of the nodes is valid and which is invalid for the schema? Explain your answer in each case.



Name: .....

Name: .....

Name: .....

Name: .....