Homework 3

Michael Clausen

Due: check on Canvas.
Group work: You may work in groups of 1-3. Include all group member names in the PDF file. You may work across sections 01 and 02. Prepare your answers as a **single PDF file**. Only one person in the group should submit to Canvas.

1. Consider the following table in a relational database storing the assignment of courses to classrooms in a university.

Course number (primary key)	Room	Department
CPSC-583	CS-110B	ComputerScience
CPSC-597	CS-110B	ComputerScience
CPSC-473	CS-406	ComputerScience

Convert the relational data into a Semantic Web representation. Use the following two properties:

```
http://example.org/is-located-in
http://example.org/is-offered-by
```

(a) Give the **triple** representation of the Semantic Web data. The first triple is already given (ignoring the full URI for subject and object).

PREFIX ex: http://example.org/

CPSC-583 ex:is-located-in CS-110B .

CPSC-583 ex: is-offered-by ComputerScience.

CPSC-597 ex: is-located-in CS-110B.

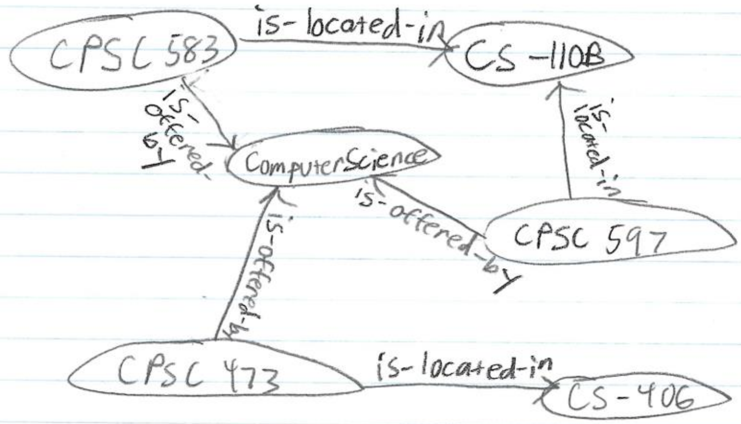
CPSC-597 ex: is-offered-by ComputerScience.

CPSC-473 ex: is-located-in CS-406.

CPSC-473 ex: is-offered-by ComputerScience.

(b) Give the **graph** representation of the Semantic Web data. The first edge is already given.

1.b)



2. Read the attached paper "Data Integration through Ontology-Based Data Access to Support Integrative Data Analysis: A Case Study of Cancer Survival" by Hansi Zhang et al. (presented at the 2017 IEEE International Conference on Bioinformatics and Biomedicine).

   a. Write a short essay (approximately 200 words) that describes

      In this essay, the authors are trying to solve the problem of integrating and querying data pertaining to cancer survival. These data come from a large variety of heterogeneous sources with varying semantics, data formats and structures, and interpretations. The authors here use semantic data integration to solve this problem. This involves developing a global ontology which can process "high level semantic queries" written in SPARQL. The authors then took relational data sets from state and federal health service sources and created semantic mappings to their global ontology. SPARQL queries are reformulated and then rewritten using the semantic mappings, which generates SQL sub-queries for the relational data bases. The data base queries are translated with the semantic mappings and then added to the pipelined integrated data set, which is sent to the user (Zhang et al., 2018).

      The author's Semantic Web-based solution is impressive. It provides a large amount of flexibility in dealing with heterogeneous data sources and should provide end users with a powerful querying tool. However, it does seem somewhat labor intensive when it comes to adding new data types and objects to the Semantic Web. Since the authors' goal is to generate statistical models, it would be interesting to see the results of developing graph neural networks for this purpose would be. This way, the model could learn the important data relations for you (Zhang et al., 2018).

References

Zhang, H., Guo, Y., Li, Q., George, T. J., Shenkman, E., Modave, F., & Bian, J. (2018). An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival. BMC Medical Informatics and Decision Making, 18(Suppl 2), 41–41. https://doi.org/10.1186/s12911-018-0636-4

   b. Figure 2 in the paper is a fragment of the ontology they developed. Using only this information, write a SPARQL query to retrieve a list of all patients who have lung cancer and who live in a county with a high rate of smoking (ocrv:avg_smoke > 0.2). Hint: you can look at some of the example queries in Tables 4-7.

```
PREFIX :http://www.semanticweb.org/ontologies/OCRV#

SELECT ?p ?county ?rate

where

{

?p a ocrv:patient.

?county a ocrv:county.

?p ocrv:has_disease ocrv:lung_cancer.

?p ocrv:lives_in ?county.

?county ocrv:avg_smoke ?rate.

FILTER(?rate > .2).

}
```

3. Write SPARQL queries to list:

   a. All English language films that starred *Irrfan Khan*.

      SELECT ?film

      WHERE

      {

      ?film a dbo:Film.

      ?film dbpedia2:starring "Irrfan Khan"@en.

      ?film dbpedia2:language "English"@en.

      }

   b. All English language films that starred *Irrfan Khan* and *Angelina Jolie*.

      SELECT ?film

      WHERE

      {

      ?film a dbo:Film.

      ?film dbpedia2:starring "Irrfan Khan"@en.

      ?film dbpedia2:starring "Angelina Jolie"@en.

      ?film dbpedia2:language "English"@en.

      }

   c. All English language films set in both *India* and *China*

      SELECT ?film

      WHERE

      {

?film a dbo:Film.

?film dbpedia2:country "China"@en.

?film dbpedia2:country "India"@en.

}

d. All actors who have acted in both “*Corpse Bride*” and “*Alice in Wonderland*” (2010 film)

SELECT ?film ?actor

WHERE

{

?film a dbo:Film.

?film rdfs:label "Alice in Wonderland (2010 film)"@en.

:Corpse_Bride dbpedia2:starring ?actor.

?film dbpedia2:starring ?actor.

}

For each query, give the SPARQL code and the output.

Hints: The SPARQL queries that were demonstrated in class are in the sparql_examples.txt file on Canvas. These queries can be executed on the SPARQL endpoint for the dbpedia dataset: http://dbpedia.org/snorql/. Films are of type dbo:Film, other useful properties are dbpedia2:language, dbpedia2:starring, dbpedia:country. You can do partial string matching like so: FILTER contains(?actorname, "Irrfan"). You can check the language encoding of a string like so: (lang(?actorname) = 'en').

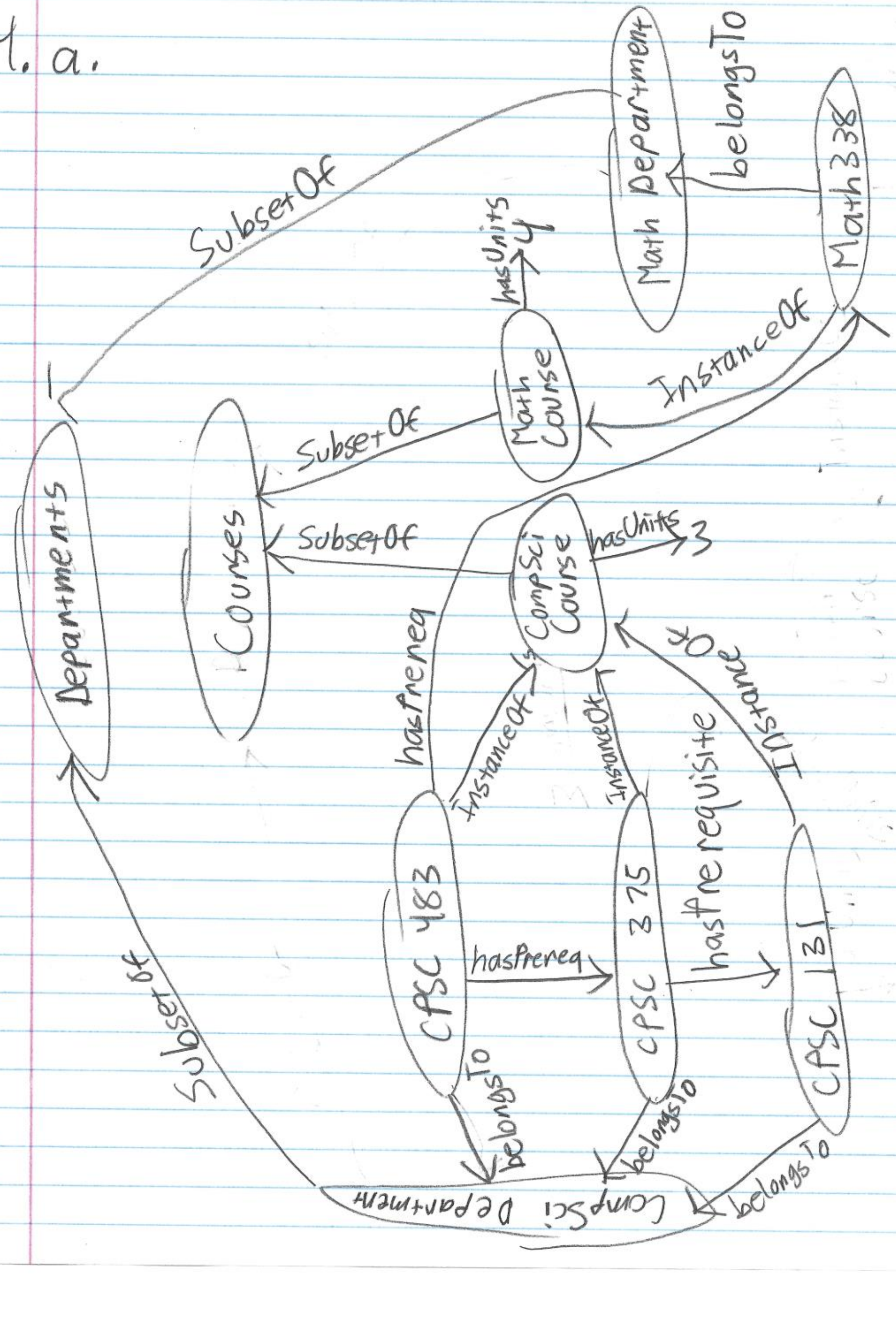
4. Consider the following facts.

   CPSC131, CPSC375, CPSC483 are Computer Science courses. MATH338 is a Math course. CPSC483 has CPSC375 and MATH338 as prerequisites. CPSC375 has CPSC131 as a prerequisite. All courses are 3 units except Math courses which are 4 units.

Every course belongs to a department. Computer Science Department and Mathematics Department are two departments.

a. Represent the above facts as a semantic network (draw the network). Use the following properties/relationships: instanceOf, subsetOf, hasPrerequisite, belongsTo, hasUnits.

4. a.


Departments
Courses
Math Department
Math 338
Math Course
CompSci Course
CPSC 483
CPSC 375
CPSC 131
CompSci Department
SubsetOf
SubsetOf
SubsetOf
Subset Of
belongsTo
belongsTo
belongsTo
belongsTo
hasUnits
4
hasUnits
3
InstanceOf
InstanceOf
InstanceOf
InstanceOf
hasPrereq
hasPrereq
hasPrerequisite

b. Using default reasoning, answer the following queries or state that they cannot be answered.
    i. How many units is CPSC131?
       3
       InstanceOf(CPSC131, CompSci Course) -> hasUnits(CompSci Course, 3)

    ii. How many units is MATH338?
       4
       InstanceOf(Math338, Math Course) -> hasUnits(Math Course, 4)

    iii. Is CPSC375 a prerequisite of CPSC483?
       True
       hasPrerequisite(CPSC483, CPSC375)

    iv. Is CPSC131 a prerequisite of CPSC483?
       True
       hasPrerequisite(CPSC483, CPSC375) -> hasPrerequisite(CPSC375, CPSC131)

    v. What are the prerequisites of CPSC483?
       CPSC 483, Math 338, CPSC 131

       hasPrerequisite(CPSC483, CPSC375) -> hasPrerequisite(CPSC483, CPSC131), hasPrerequisite(CPSC483, Math338)

    vi. Does CPSC131 belong to Computer Science Department?

       True

       belongsTo(CPSC131, CompSci Department)