CS5560 Knowledge Discovery and Management

Problem Set 7 & 8

Submission Deadline: July 28, 2017

ttps://goo.gl/forms/aTXnl4oRHMdS8j1L2

Name: Megha Nagabhushan

Class ID: 15

References

I. Logical knowledge representation

First Order Logic Reference: http://pages.cs.wisc.edu/~dyer/cs540/notes/fopc.html

1) Let us define the statements as follows:
   - G(x): "x is a giraffe"
   - F(x): "x is 15 feet or higher,"
   - Z(x): "x is animal in this zoo"
   - M(x): "x belongs to me"

   Express each of the following statements in First-Order Logic using G(x), F(x), Z(x), and M(x).

   a) Nothing, except giraffes, can be 15 feet or higher;
   b) There is no animal in this zoo that does not belong to me;
   c) I have no animals less than 15 feet high.
   d) All animals in this zoo are giraffes.

2) Which of the following are semantically and syntactically correct translations of "No dog bites a child of its owner"? Justify your answer

   a) $\forall$ x Dog(x) $\Rightarrow \neg$Bites(x, Child(Owner(x)))
   b) $\neg\exists$ x, y Dog(x) $\wedge$ Child(y, Owner(x)) $\wedge$ Bites(x, y)
   c) $\forall$ x Dog(x) $\Rightarrow$ ($\forall$ y Child(y, Owner(x)) $\Rightarrow \neg$Bites(x, y))
   d) $\neg\exists$ x Dog(x) $\Rightarrow$ ($\exists$ y Child(y, Owner(x)) $\wedge$ Bites(x, y))

3) For each of the following queries, describe each using Description Logic
Reference: http://www.inf.ed.ac.uk/teaching/courses/kmm/PDF/L3-L4-DL.pdf

   a) Define a person is Vegan
   b) Define a person is Vegetarian
   c) Define a person is Omnivore

II. SPARQL

Reference: https://www.w3.org/2009/Talks/0615-qbe/

Design a SPARQL query for following queries and show an expected output.

Query #1: Multiple triple patterns: property retrieval
Find me all the people in Tim Berners-Lee's FOAF file that have names and email addresses. Return each person's URI, name, and email address.

Query #2: Multiple triple patterns: traversing a graph
Find me the homepage of anyone known by Tim Berners-Lee.

Query #3: Basic SPARQL filters
Find me all landlocked countries with a population greater than 15 million.

Query #4: Finding artists' info
Find all Jamendo artists along with their image, home page, and the location they're near, if any.

Query #5. Design your own query

III. SWRL

References:
https://www.w3.org/Submission/SWRL/
https://dior.ics.muni.cz/~makub/owl/

Design SWRL rules for the following cases

Rule #1: design hasUncle property using hasParent and hasBrother properties
Rule #2: an individual X from the Person class, which has parents Y and Z such that Y has spouse Z, belongs to a new class ChildOfMarriedParents.
Rue #3: persons who have age higher than 18 are adults.
Rue #4: Compute the person's born in year
Rule #5: Compute the person's age in years
Rule #6: Design your own rule

① a. $\forall x(\neg G(x) \longrightarrow \neg F(x))$

b. $\forall x(Z(x) \longrightarrow M(x))$

c. $\forall x(M(x) \longrightarrow F(x))$

d. $\forall x(Z(x) \longrightarrow G(x))$

② b and c. are semantically and syntactically correct translations of "No dog bites a child of its owner".

a and d are incorrect because -

a implies that dogs do not bite dogs and child of owner of dog.

d implies that dogs bite the children of the owner.

③ Vegan: People who do not eat animal products

$\boxed{\forall \text{eats} : \neg \text{ Animal Products}}$

Vegetarian: People who do not eat animals

$\boxed{\forall \text{eats} \neg \text{ Animal}}$

Omnivore: People/Animals who eat both plant and animals

$\boxed{\exists \text{eats. Animals}}$

II ①

```
PREFIX foaf : <http://xmlns.com/foaf/0.1/>
SELECT *
WHERE {
        ?person foaf:name ?name.
        ?person foaf:mbox ?email.
   }
```

②

```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
PREFIX card :<http://www.w3.org/People/Berners-Lee/card#>
SELECT ?homepage
FROM <http://www.w3.org/People/Berners-Lee/card>
WHERE {
      card:i foaf:knows ?known.
      ?known foaf:homepage ?homepage.
 }
```

③

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>

 SELECT ?country-name ?population.
  WHERE {
      ?country a type:LandlockedCountries ;
                rdfs:label ?country-name ;
                prop:populationEstimate ?population
          FILTER (?population > 15000000)
   }
```

④

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX foaf: <http://xmlns.com/foaf/0.1>
SELECT  ?name ?img ?hp ?loc
WHERE {
      ?a  a: mo:MusicArtist;
          foaf:name  ?name;
          foaf:img  ?img;
          foaf:homepage  ?hp;
          foaf:based_near  ?loc.
  }
```

⑤

```
PREFIX  space:<http://purl.org/net/schemas/space/>
PREFIX   xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT  *
{  ? launch  space:launched  ? date
 FILTER (
      ?date >"1968-10-1"^^xsd:date &&
      ?date < "1968-10-30"^^ xsd:date
    )
}
```

This query finds all launches in October 1968

III

Rule #1:

hasParent (?x1,?x2) ∧ hasBrother (?x2, ?x3)
=> hasUncle (?x1, ?x3)

Rule # 2:

Person(?x), hasParent (?x, ?y), hasParent (?x, ?z),
hasSpouse (?y, ?z) -> ChildOfMarriedParents (?x)

Rule # 3:

Person (?p), hasAge(?p, ?age), swrlb: greaterThan
(?age, 18) ⟶ Adult (?p)

Rule# 4:

Person(?p), bornOnDate (?p, ?date), xsd:date(?date),
swrlb: date(?date, ?year, ?month, ?day ; ?timezone)
-> bornInYear(?p, ?year)

Rule #5:

Person(?p), bornInYear (?p, ?year), my: thisYear (?nowyear),
swrlb: subtract (?age, ?nowyear, ?year) ->hasAge(?p, ?age)

Rule #6: The following rule uses data range restriction.
Data range restrittion is satisfied when the ?age
variable has an integer value between 18 and 65
inclusive.

Person (?p), integer [>=18, <=65] (?age), hasAge (?p, ?age) ⟹ hasDriverAge (?p, true)