CS5560 Knowledge Discovery and Management

Problem Set 7 & 8

Submission Deadline: July 28, 2017

ttps://goo.gl/forms/aTXnl4oRHMdS8j1L2

Name: Syed Moin
Class ID: 28

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

I) Logical Knowledge Representation:

First order logic (FOL or FOPC) :

User defines these primitives:

1) Constant symbols (ie, "individuals in the world")
2) Function symbols (mapping individuals to individuals)
3) predicate symbols (mapping from individuals to truth values)

FOL supplies Ex: x, y these primitives

1) Variable symbols Ex: x, y
2) connectives not ($\neg$), and ($\wedge$) or ($\vee$), implies ($\Rightarrow$), if and only if ($\Leftrightarrow$)
3) Quantifies: Universal (A) and Existential (E)

(1) Possible translation for the given statements are.

$\forall x\ (\neg G(x) \rightarrow \neg F(x))$ or $\forall x\ (F(x) \rightarrow G(x))$

$\neg \exists x (Z(x) \wedge \rightarrow M(x))$ or $\forall x (Z(x) \rightarrow M(x))$

$\forall x\ (M(x) \rightarrow F(x))$

$\forall \neg (Z(x) \rightarrow G(x))$

(2) Syntactic Analysis:

The goal of Syntactic analysis is to determine whether the text string on input is a sentence in the given natural language.

Semantic Analysis:-

Semantic and pragmatic analysis make up the most complex phase of language processing as they build up on results of all the mentioned disciplines.

a) $\forall x \; Dog(x) \Rightarrow \neg Bites(x, childlowner(x))$

No dog bites dogs and owner of children.

b) $\neg \exists x,y \; Dog(x) \wedge child(y, owner(x)) \wedge Bites(x, y)$

No dog bites owners children.

c) $\forall x \; Dog(x) \Rightarrow (\forall y child(y, owner(x)) \Rightarrow \neg Bites(x, y)$

All dog do not bite their children of owner.

d) $\neg \exists x \; Dog(x) \Rightarrow (\exists y \; child(y, owner(x)) \wedge Bites(x, y)$

Dog bite the children of owners.

Therefore, the correct translations are (b) and (c)

(3) Description logic:- Description logic allows formal concept definitions that can be reasoned about to be expressed. It is an important element of the semantic web.

a) Define a person is vegan

People who does not eat or use animal products

$\forall$ eats $\neg$ Animal products.

b) Define a person is vegetarian

people who does not eat animal products

$\forall$ eats $\neg$ Animal

c) Define a person is omnivore

Animal/person eats food of both plant and animal

$\exists$ eats Animal.

II SPARQL:

Query 1: Multiple triple patterns: property retrieval

```
Prefix foaf : <http://xmlns.com/foaf/0.1>
  Select *
    where {
          ?person foaf : name ?name
          ?person foaf : mbox ?email
```

Expected output: person name email

<http://www.w3.org/people/Berners_Lee/"Amy Vanderbilt"<mailto:amy@w3.org>

<http://www.w3.org/people/Berners_Lee/card#ds>"Dean Jackson"
<mailto:dean@w3.org>

<http://www.w3.org/people/Berners_Lee/card#edd>"Dean Edd"
<mailto:edd@usefulinc.com>

Query 2: Multiple triple patterns : traversing a graph

```
Prefix foaf : <http://xmlns.com/foaf/0.1>
Prefix card : <http://www.w3.org/people/Berners-Lee/card#>
    Select ? home page
    from <http://www.w3.org/people/Berners-Lee/card>
    where {
        card: 1 foaf :knows ? known
            ? known foaf : home page ? home page
    }
```

Expected output:

```
<http://purl.org/net/leo/>.
<http://www.mellon.org/about-foundation/staff/programs-area-staff/rafunchs>.
<http://www.johnseelybrown.com/>
<http://heddley.com/lead>
```

Query 3: Basic SPARQL filters:

```
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 {

∀ eats ⊤ Animals products

b) Define a person is vegetarian

people who does not eat animal

¬∃ eat ⊤ Animal.

c) Define a person is omnivore.

Animal/person eats food of both plant and Animal

∃ eats Animal

II SPARQL:

Query 1: Multiple triple patterns: property retrieval

```
PREFIX foaf : <http://xmlns.com/foaf/0.1>

Select *
where {
        ? person a type : Landlocked countries ;
    rdfs : label ? country-name ;
    pop : populationEstimate ? population
    FILTER ( ? population > 150000000 )
    }
```

Expected output:

Country name	population
Afghanistan	31889923
Afghanistan	31889923
⋮	
Ethopia	75067000
Ethpia	75067000
⋮	⋮

Query 4: finding artists info

```
Prefix MO: <http://purl.org/ontology/mo/>.
prefix foaf: <http://xmlns.com/foaf/0.1/>
  select ?name ?img ?hp ?loc
   where {
     ?a a MO:Music Artist;
        foaf:name ?name;
```

```
foaf:img ?img;
foaf:homepage ?hp;
foaf:based_near ?loc
```

wrong way

```
optional { ?a foaf:img ?img }
optional { ?a foaf:home page ?hp }
optional { ?a foaf:based_near ?loc }
```

Right way

Expected output

name img

" Cicada " ^^ xsd:string http://img.jamendo.com/artists/h/Patrickman.jpg

http://www.loc.circada.org.st

http://sws.geonames.loc.org/3031234

Query 5: Design your own query.

Asking Question → Is the Amazon river longer than the Nile River?

PREFIX : <http://dbpedia.org/property/>

ASK

<http://dbpedia.org/resource/Amazon_River> prop:length ?amazon

<http://dbpedia.org/resource/Nile> prop:length ?nile.

FILTER (?amazon > ?nile)

}

Expected output: <?xml version="1.0"?>

<sparql xmlns="http://www.w3.org/2005/sparql-results#"

xmlns:xsi="http://www.w3.org/2001/sw/DataAccess/rf1/result?xsd">

<head></head>

<boolean>true</boolean>

</sparql>

III SWRL:

A semantic web rule language combining OWL & RuleML

Rule #1: design has uncle property using has parent & has brother properties

has parent $(?x_1, ?x_2)$ ∧ has Brother $(?x_2, ?x_3) \Rightarrow$ has uncle $(?x_1, ?x_3)$

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 child of married parents.

person(?x), has parent (?x, ?y), has parent (?x, ?z), has spouse (?y, ?z)

child of married parents (?x)

Rule #3: persons who have age higher than 18 are adults

person (?p), has Age (?p, ?age), swrlb: greater than (?age, 18) → Adult (?p)

Rule #4: compute the person's born in year

person (?p), born on date (?p, ?date), xsd: date (?date), swrldf: date (?date, ?year, ?month, ?date, ?timezone) → born In year (?p, ?year)

Rule #6: design your own rule

→ design has on property using has child and Man properties

has child (?x, ?y) ∧ man (?y) ⇒ has son (?x, ?y)