

CSE 7320/5320

Assignment 7: Knowledge Rep

Due: Sunday April 7 midnight

Download the semweb.jar file from Canvas.

PART A. Just the FACTS:

Convert the following data to turtle/n3. Use anonymous nodes for the people and places since we do not have a standard URL for them. Explore schema.org for relevant terms and attributes.

Name your file a7data.txt

DATA:

A man called Zippy lives in campground in Boston. His favorite team is the Boston Celtics. He likes to drink beer and tequila.(see schema.org for Sports teams)

A man called Rollo lives in Dallas. His favorite team is the Dallas Mavericks. He has brown hair and likes to drink wine and tequila. He knows Marla and Zippy and Zonker. He works as a 'Manager' at an Internet Café on Skillman Avenue.

A woman called Marla lives in Dallas. She has red hair and drinks coffee. She works as a 'Stylist' at the 'Cut Me Up' hair salon in Dallas.

A man called Ralph is the brother of Marla. He works as a 'Shoe Sizer' at the 'Big Bowl' bowling alley.

A man named "BigZ" drinks tea and tequila.

A man named Zonker knows Rollo. He created the video game 'How High Can You Jump'. He works at the 'Big Teepee' Casino in Dallas as a 'Dealer'. Zonker is the brother of Marla.

PART B. Ontology

Use RDFS and OWL to create an ontology file about the facts described above. Name the file a7onto.txt

- For any classes in schema.org – check their class hierarchy and represent the hierarchy as `rdfs:subClassOf`.
- Also, investigate schema.org for opportunities to use `rdfs:domain` and `rdfs:range` in your ontology.
- All classes in schema.org are considered subclasses of `Thing`.

Describe the following in your ontology file. Most can be done use OWL constructs.

- If x knows y, then infer y knows x

- If x works at y, then y employs x.
- If x has a sister y, then y has a brother x.
- If x has a brother y, and y has a brother z, then x has a brother z.
- If x has brother or sister, then x has a sibling (hint: rdfs:subPropertyOf)
- Wine is a kind of 'Alcoholic Beverage' (hint: rdfs:subClassOf)
- Beer is a kind of 'Alcoholic Beverage'
- Tequila is a kind of 'Alcoholic Beverage'
- If x knows y, then both x and y are foaf:Person(s)
- If x has a foaf:gender property, then x is a foaf:Person.
- BigZ and Zonker are the same person.

Test your facts and ontology by executing the semweb.jar code.

At a command prompt:

- `%java -jar semweb.jar -d a7data.txt -o a7onto.txt`
- Capture the expanded triples.

Part C. Rules

- Create one or more rules that result in new triples that can be used in a SPARQL query.
- Example: Any Person who works in a Casino and drinks alcoholic beverages is dexterous (good with their hands).
- Feel free to add your own rules and to add new facts.
- Test with: `%java -jar semweb.jar -d a7data.txt -o a7onto.txt -r a7rules.txt`

Part D. SPARQL

Write some SPARQL query that showoff your ontology and rules.

- Find all the siblings of a Person named Marla. (sparql1.txt)
- Find anyone employed by a Casino who has a sister named Marla. (sparql2.txt)
- Create a query that is based on one or more of your rules (sparql3.txt)
- Execute your SPARQL query using:
- `%java -jar semweb.jar -d a7data.txt -o a7onto.txt -r rules.txt -s a7sparql1.txt`
- `%java -jar semweb.jar -d a7data.txt -o a7onto.txt -r rules.txt -s a7sparql2.txt`

Submit:

A text file or PDF that contains:

- Data file from Part A.
- Onto file from Part B.
- Rules file from Part C.
- Sparql file from Part D.

Captured output including:

- Triples after inferencing
- Results of your SPARQL queries

Example:

rollo2.txt

```
@prefix :      <http://www.codesupreme.com/#>.
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .

:wally :residesIn :Dallas .
:wally :age 23 .
:wally foaf:knows :marla .

:marla a :Person .
:marla :age 25 .
:marla :livesIn :California .

:rollo rdf:type :Person .
:rollo :age 22 .
:rollo :livesIn :Texas .
:rollo :loves :marla .
```

rollo.owl.txt

```
@prefix :      <http://www.codesupreme.com/#> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .

:livesIn owl:equivalentProperty :residentOf.
foaf:knows  rdf:type owl:SymmetricProperty .
:ancestorOf rdf:type owl:TransitiveProperty .
:roro owl:sameAs :rollo.
:livesIn owl:inverseOf :homeOf.
foaf:knows rdfs:range foaf:Person .
foaf:knows rdfs:domain foaf:Person .
```

rules.txt

```
# Example rule file
@prefix :      <http://www.codesupreme.com/#>.

[rule1: (?u :loves :marla )  -> (?u :loves :food)]
[rule2: (?u :loves :food )    -> (?u :loves :mustard)]
[rule3: (?x :loves :mustard) -> print('Mustard World Located in Rule set')
(?x rdf:type :MustardEatingPerson)]

# The "IF" part of a rule MUST use rdf:type rather than n3 shorthand "a"
# However, it is OK to use "a" in the "THEN" part since the THEN part will
# become part of a model which will do the translation from 'a' to
# 'rdf:type'

[rule4: (?x rdf:type :Person) -> (?x rdf:type :Human) ]

# Rule to print output (for debugging)
[rule5: (?x rdf:type :Person) -> print(?x, ' is a Person')  ]

#A rule with two parts (AND) -- then
[rule6: (?x :livesIn :Texas) ,(?x rdf:type :Person) -> (?x rdf:type :Texan)]
```

SPARQL

```
prefix : <http://www.codesupreme.com/#>
prefix foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?y
WHERE {
    :rollo rdf:type ?y .
}

>java -jar semweb.jar -d rollo2.n3 -q wallyq.sparql -o rollo.owl.txt
-r rules.txt -s wallyq.txt > results.txt

RESULTS:
q : [wallyq.sparql]
r : [rules.txt]
s : [wallyq.txt]
d : [rollo2.n3]
```

o : [rollo.owl.txt]

Number of parameters = 5

Your ontology file must contain '@prefix owl:

<http://www.w3.org/2002/07/owl#> .'

Success Loading Ontology File:

--- Your Triples Data After Inferencing and Rules ---

```
:rollo a          :MustardEatingPerson , :Texan , :Human , :Person
, owl:Thing , rdfs:Resource ;
    :age          22 ;
    :livesIn      :Texas ;
    :loves        :mustard , :food , :marla ;
    :residentOf   :Texas ;
    owl:sameAs
        :roro , :rollo .
```

```
:marla a          :Human , :Person , rdfs:Resource , foaf:Person ;
    :age          25 ;
    :livesIn      :California ;
    :residentOf   :California ;
    foaf:knows    :wally .
```

```
:roro a          :MustardEatingPerson , :Texan , :Human ,
owl:Thing , rdfs:Resource , :Person ;
    :age          22 ;
    :livesIn      :Texas ;
    :loves        :mustard , :food , :marla ;
    :residentOf   :Texas ;
    owl:sameAs
        :roro , :rollo .
```

```
:age a          rdf:Property , rdfs:Resource ;
    rdfs:subPropertyOf :age .
```

```
:residesIn a      rdf:Property , rdfs:Resource ;
    rdfs:subPropertyOf :residesIn .
```

```
:livesIn a        rdf:Property , rdfs:Resource ;
    rdfs:subPropertyOf :residentOf , :livesIn ;
    owl:equivalentProperty
        :livesIn , :residentOf ;
```

```

    owl:inverseOf
        :homeOf .

:residentOf a          rdf:Property , rdfs:Resource ;
    rdfs:subPropertyOf :livesIn , :residentOf ;
    owl:equivalentProperty
        :livesIn , :residentOf .

:homeOf owl:inverseOf
        :livesIn .

:loves a          rdf:Property , rdfs:Resource ;
    rdfs:subPropertyOf :loves .

:wally a          rdfs:Resource , foaf:Person ;
    :age          23 ;
    :residesIn    :Dallas ;
    foaf:knows    :marla .

:ancestorOf a          owl:TransitiveProperty , rdf:Property ,
    owl:ObjectProperty , rdfs:Resource ;
    rdfs:subPropertyOf :ancestorOf .

:Texas :homeOf :rollo , :roro .

:California :homeOf :marla .

:Person a          rdfs:Class , rdfs:Resource .

```

-----RESULTS of SPARQL QUERY

```

-----
| y |
=====
| <http://www.codesupreme.com/#MustardEatingPerson> |
| <http://www.codesupreme.com/#Texan> |
| <http://www.codesupreme.com/#Human> |
| <http://www.codesupreme.com/#Person> |
| <http://www.w3.org/2002/07/owl#Thing> |
| <http://www.w3.org/2000/01/rdf-schema#Resource> |
-----

```