Knowledge Engineering and Semantic Web

Exercise Sheet: 4

Will be discussed on: May 23, 2023

Leibniz Universität Hannover

TUTORS:
Yaser Jaradeh, Hassan Hussien, and some other ORKG members

QUESTIONS: Please don't hesitate to ask any questions. Questions help you and your peers.
PRINT: Please consider the environment before printing the exercise.

1 Review questions

1. Semantics and Syntax : which statement is correct?
   (a) **Syntax:** meaning of the character strings
   (b) **Semantics:** meaning of the character strings
2. Implicit Knowledge: which statement is correct?

```
ex:ExampleBook rdf:type ex:TextBook .
ex:TextBook rdfs:subClassOf ex:Book .
ex:PrintMedia rdfs:subClassOf ex:Media .
```

   (a) `ex:TextBook rdf:type ex:ExampleBook .`
   (b) `ex:TextBook rdf:type ex:PrintMedia .`
   (c) `ex:ExampleBook rdf:type ex:Book.`
3. rdfs:subClassOf "characteristics"
   (a) symmetric
   (b) transitive
   (c) reflexive

2 Semantic Entailment Rules

Write down the semantic entailment rules.
Note:
if $a\ \eta\ b \in K$ and $b\ \eta\ c \in K$ then $K \leftarrow K \cup \{a\ \eta\ c\}$
also written as : $\frac{a\ \eta\ b \quad b\ \eta\ b}{a\ \eta\ c}$

Notation Triples : < subject predicate object >

a and b $\widehat{=}$ any URI (predicates)

$\_:n$ $\widehat{=}$ blank node ID

u and v $\widehat{=}$ any URI or blank node ID for (subject)

x and y $\widehat{=}$ any URI, or blank node ID or literals (object)

l $\widehat{=}$ any literal

Simple Entailment Rules

Rule	Formula
Simple Entailment 1	
Simple Entailment 2	

Explain in own words the two rules.
Rule 1:

Rule 2:

RDF Entailment Rules

Rule	Formula
RDF axioms	
Literal Grounding	
RDF Rule 1	
RDF Rule 2	

RDFS Entailment Rules

Rule	Formula
RDFS axioms	
RDFS Rule 1	
RDFS Rule 2 (domain)	
RDFS Rule 2 (range)	
RDFS Rule 4a	
RDFS Rule 4b	
RDFS Rule 5	
RDFS Rule 6	
RDFS Rule 7	
RDFS Rule 8	
RDFS Rule 9	
RDFS Rule 10	
RDFS Rule 11	
RDFS Rule 12	
RDFS Rule 13	

3 Inference

1. Hierarchy of properties: Select the correct inferences among the following ones.

   a)
   ```
   :a :p1 :b .
   :a :p2 :c .
   ->
   :b rdfs:subPropertyOf :c .
   ```

   b)
   ```
   :a :p1 :b .
   :b rdf:type :C .
   ->
   :p1 rdfs:range :C .
   ```

   c)
   ```
   :a :p1 :b .
   :p2 rdfs:subPropertyOf :p1 .
   ->
   :a :p2 :b .
   ```

   d)
   ```
   :p1 rdfs:subPropertyOf :p2 .
   :a :p1 :b .
   ->
   :a :p2 :b .
   ```

2. Hierarchy of Classes: Select the correct inferences among the following ones.

   a)
   ```
   :A rdfs:subClassOf :B .
   :c rdf:type :A .
   ->
   :c rdf:type :B .
   ```

   b)
   ```
   :a :p1 :b .
   :a :p2 :c .
   :b rdf:type :B .
   :c rdf:type :C .
   :B rdfs:subClassOf :C .
   ->
   :p1 rdfs:subPropertyOf :p2 .
   ```

   c)
   ```
   :p1 rdfs:domain :A .
   :p1 rdfs:range :C .
   :p2 rdfs:domain :B .
   :p2 rdfs:range :D .
   :p1 rdfs:subPropertyOf :p2 .
   ->
   :A rdfs:subClassOf :B .
   :C rdfs:subClassOf :D .
   ```

   d)
   ```
   :a :p1 :b .
   :p2 rdfs:domain :C .
   :p1 rdfs:subPropertyOf :p2.
   ->
   :a rdf:type :C .
   ```

3. Equivalence of Classes: Select the correct inferences among the following ones.

   a)
   ```
   :A rdfs:subClassOf :B .
   :B rdfs:subClassOf :C .
   :C rdfs:subClassOf :D .
   :D rdfs:subClassOf :A .
   ->
   :A , :B , :C , :D
   are equivalent classes.
   ```

   b)
   ```
   :A rdfs:subClassOf :B .
   :B rdfs:subClassOf :C .
   :c rdf:type :A .
   ->
   :c rdf:type :C .
   ```

   c)
   ```
   :A rdfs:subClassOf :B .
   :B rdfs:subClassOf :A .
   :c rdf:type :A .
   :d rdf:type :A .
   ->
   :c and :d are equivalent.
   ```

   d)
   ```
   :p1 rdfs:subPropertyOf :p2 .
   :p2 rdfs:subPropertyOf :p1 .
   :p1 rdfs:range  :B ;
       rdfs:domain :A .
   :p2 rdfs:range  :D .
   :p2 rdfs:domain :C .
   ->
   :A is equal to :C and :B is equal to :D.
   ```

4 Consider the following statements:

a) Represent them in RDF Turtle serialization.

b) Select the correct ones.

1. $< rdfs : subClassOf^I, rdfs : Resource^I > \in I_{EXT}(rdfs : domain^I)$.
2. $< rdf : List^I, rdf : rest^I > \in I_{EXT}(rdfs : domain^I)$.
3. $I_{CEXT}(rdfs : Class^I) \subseteq I_{CEXT}(rdfs : Resource^I)$.
4. $< rdfs : domain^I, rdf : Property^I > \in I_{EXT}(rdf : type^I)$.
5. If $< x, y > \in I_{EXT}(rdfs : domain^I)$ and $< u, v > \in I_{EXT}(x) \rightarrow u \in I_{CEXT}(x)$.

5 For the following knowledge base, indicate which statement can be entailed. Prove the true answers with proof-theoretic semantics.

```
@prefix ex:   <http://example.org> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .



ex:dog      rdfs:subClassOf   ex:animal .
ex:horse    rdfs:subClassOf   ex:creature .
ex:person   rdfs:subClassOf   ex:creature .


ex:isEnemyOf     rdfs:subPropertyOf   ex:knows .
ex:isEnemyOf     rdfs:domain          ex:person;
                 rdfs:range           ex:person .
ex:isFriendOf    rdfs:subPropertyOf   ex:knows .




ex:LuckLuke      a   ex:person .
ex:JollyJumper   a   ex:horse .
ex:Rantanplan    a   ex:dog .


ex:LuckyLuke     ex:isFriendOf   ex:JollyJumper .
ex:JollyJumper   ex:isFriendOf   ex:Rantanplan .
ex:LuckyLuke     ex:isEnemyOf   exJoeDalton .
```

Statements:

1. `ex:Rantanplan a ex:creature.`
2. `ex:Rantanplan ex:isFriendOf ex:JollyJumper.`
3. `ex:LuckyLuke ex:isFriendOf ex:RantanPlan.`
4. `ex:LuckyLuke ex:knows ex:JoeDalton.`
5. `ex:JoeDalton ex:isEnemyOf ex:LuckyLuke.`
6. `ex:JoeDalton a ex:creature.`