SEMANTIC WEB AND INFORMATION EXTRACTION TECHNOLOGIES

Spring 2023 Assignment-1

SHUBHIKA GARG

1. **“http://www.semanticweb.org/websem-spring/ontologies/2023/2/GARG-cycling”** can work as a good namespace URI for the Cycling domain. For a unique namespace prefix, we could use "**wsc**" to represent the URI. I chose this as we have to avoid name collisions.

2. Let **C1** be **Race** class, **C2** be **OneDayRace** class and **C3** be **SeveralStagesRace** class
   C1 ≡ C2 ⊔ C3
   Here, the equivalence symbol "≡" indicates that the left-hand side and right-hand side are logically equivalent. The symbol "⊔" represents the "union" of the two classes.
   Also, I defined **Race class** as a **disjoint Union Of OneDayRace, SeveralStagesRaces**

3. I created an Object property **composedOf** with domains: **SeveralStagesRaces** and Ranges: **Stages.**

4. I created the classes **MountainStage**, **FlatStage** and **TimeTrial** as the **subclasses** of **Stages** and **IndividualTimeTrial** and **TeamTimeTrial** as **subclasses** of **TimeTrial**. A TimeTrial is either a IndividualTimeTrial or a TeamTimeTrial.

5. **Prologue ≡ Stages and (hasStageNumber exactly 1 xsd:positiveInteger)**
   The above expression states that there exists a "SeveralStagesRace" that has "Prologue" as its first stage, and the "≡" symbol denotes equivalence or strict equality. The data property "**hasStageNumber**" has been defined with a range of a Positive Integer and a domain of SeveralStagesRace

6. I created classes “**Spectactor**”, “**RacePerson**” or “**TeamPerson**” as the subclasses of the class “**Person**”.
   TeamPerson is Equivalent To: Doctor or Director or RaceCyclist
   RaceCyclist is Equivalent To: Rider or Sprinter or Climber
   I have added three Data properties: **name**, **age** and **nationality** with Domains: **Person**
   Furthermore, another Object property **participatesIn** has been added with Domains: **Person** and Ranges: **Race**.

Below are the differences between Object and Datatype properties:

SERIAL NUMBER	OBJECT PROPERTIES	DATATYPE PROPERTIES
1.	They link individuals to other individuals.	They link individuals to data values.
2.	They can be transitive, symmetric, or reflexive.	They do not have any of these characteristics.
3.	They are usually used to define more complex relationships between individuals.	They are used to define basic attributes or characteristics of individuals.
4.	They relate to relationships or connections between individuals.	They describe some aspect of an individual, such as name, age, height or weight.

Example:

Suppose in a zoo Ontology we have an object property "hasHabitat", which links an animal to the habitat where it lives. *Lion* <u>*hasHabitat*</u> *savanna.*

We also have a datatype property age and weight which specifies the age and weight of an animal. *Lion* <u>*hasAge*</u> *15 years and* <u>*hasWeight*</u> *88 kg.*

Here, "hasHabitat" is an object property because it links an animal to a habitat, which is another individual. “hasAge” and "hasWeight" is a datatype property because they describe a numerical value that is associated with the animal.

7. "**belongsTo**" property is used to define the relationship between **Team** and its members. It is used to state that **Team** has exactly one **Doctor**, one **Director**, and ten **RaceCyclist**. In this property, I gave the cardinality of the three classes: **Doctor:1, Director:1 and RaceCyclist:10.** Also, since we are defining a Team: I used (inverse)belongsTo property.
   (inverse(belongsTo) exactly 1 Doctor) and (inverse(belongsTo) exactly 1 Director) and (inverse(belongsTo) exactly 10 RaceCyclist)

8. Defined **GoodTeam** as a subclass of **Team** equivalent to the class expression:
   (inverse(belongsTo) min 3 Climber) and (inverse(belongsTo) exactly 1 Rider)
   The above statement means the GoodTeam has exactly 1 Rider and at least (a minimum of) 3 Climber.

9. I did direct import of **http://xmlns.com/foaf/0.1/** to my Ontology. Then, I added **foaf:Organization** and its subclass **foaf:Team** and **foaf:Person** equivalent to **Person**. Furthermore, I mapped the **name** data property with **foaf:name** property.

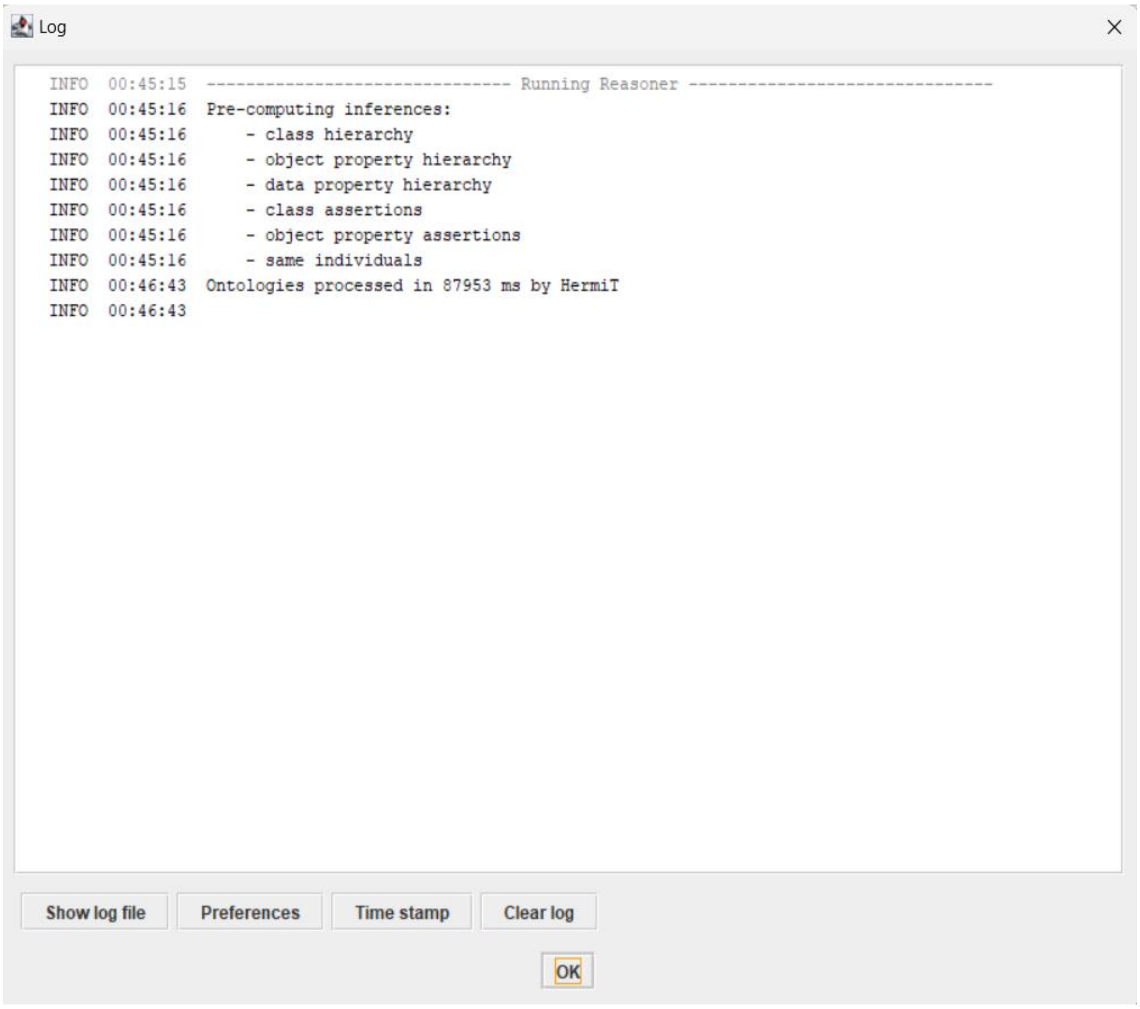
10. I added three instances in class Team (***Risers, Skema, Sky***), one instance in class SeveralStagesRace (***Tour de France)***, two instances in class Climber (***Alberto Johnson***, ***Micky Avalle***) and two instances in class Rider (***Jenny Federick, Krystle David*****).**

11. Using the LOV search function at **https://lov.linkeddata.es/dataset/lov/** we can re-use:
    - **weight** (A schema.org property) to define the weight of a Person
    - **height** (A schema.org property) to define the height of a Person
    - **foaf:Agent** An agent (e.g., person, group, software or physical artifact)
    - **foaf:gender** the gender of this Agent
    - **dbpedia-owl:HorseRider, dbpedia-owl:SpeedwayRider, dbpedia-owl:MotorcycleRider**
    - **coach** (A schema.org property) A person that acts in a coaching role for a sports team
    - **foaf:img** use when an image is particularly representative of some person.
    - **affiliation** (A schema.org property) An organization that the person is affiliated with.

12. Using the reasoner **HermiT:** After resolving all the errors, It didn’t give any errors/inferences for me.

Challenges in this assignment:

- On clicking **http://xmlns.com/foaf/spec/** it was showing *"Temporarily, please see /foaf/0.1 document."* I thought this to be some error, but after confirming with the Professor, I imported **http://xmlns.com/foaf/0.1/** to my Ontology.
  It took me sometime to figure out how to define the foaf classes like foaf:Organization to my Ontology.
- Some of the properties were not added properly like not adding DisjointUnionOf property for class Race or not defining hasStageNumber as data property for Prologue class.
- Show Inferences icon was not checked in my case, so I wasn't able to debug the errors after running the reasoner.



```
INFO  00:45:15  ------------------------------- Running Reasoner -------------------------------
INFO  00:45:16  Pre-computing inferences:
INFO  00:45:16      - class hierarchy
INFO  00:45:16      - object property hierarchy
INFO  00:45:16      - data property hierarchy
INFO  00:45:16      - class assertions
INFO  00:45:16      - object property assertions
INFO  00:45:16      - same individuals
INFO  00:46:43  Ontologies processed in 87953 ms by HermiT
INFO  00:46:43
```