<div align="center">

SEMANTIC PROJECT
ECS735P

</div>

1. Introduction

I built the Cricketer ontology

(IRI: http://www.semanticweb.org/uttkarshraj/ontologies/2022/4/proj_cricketer) which has the information about all the cricketers like: their height, deliveries, their most odi runs ,their birthplace, club number, batting and bowling side and etc. I took the information from DBpedia and this ontology can be used to query an information about a cricketer. If someone is interested about Virat Kohli, then it can be found that the number of matches he has played, his number of tests, ODI runs and club number, etc.

⇨ Cricket includes two teams with 11 players on each side. The captain who wins the toss will get to decide whether his/her team bats or bowls first. Assuming they bat first their main purpose is to score a ton of tuns and ensure the other team doesn't arrive at the score.

⇨ This sport is played all around the world at the worldwide level with three different formats – Test matches, T-20 internationals, and One-Day internationals. All the matches are played under the standards and guidelines supported by the International Cricket Council (ICC), which likewise gives match authorities to them.

⇨ We have The Cricket World Cup(tournament); it is One Day International cricket played between different nations. The game is associated by the games head(administrator) body, the ICC (International Cricket Council) with fundamentals capability gathers driving together to a final's competition.

⇨ The International Cricket Council is worldwide head body of the cricket. Addressing 106 individuals, the International Cricket Council administrates the game and works with individuals to develop the game. The ICC takes main charge to organise all the ICC events.

⇨ International Cricket Council; manages the ICC code of conduct, the DRS (Decision Review System), playing conditions and all the other important guidelines

Class	Asserted
Birthplace	N/A
Captain	N/A
Cricketer	N/A
Mostruns	N/A
MostTestRuns	N/A
Odiruns	N/A

Table 1. Asserted information for each class in the Figure Cricketer ontology.

2. Concepts and Properties

2.1. Asserted class

We have 6 main classes: Birthplace, Captain, Cricketer, Mostruns, MostTestRuns and Odiruns.

1. **Cricketer**: It's the parent class and all the Object property hierarchy and Data properties have same domain as Cricketer.

CONSTRUTCT {

```
?cricketer rdf:type cr:Cricketer .
      ?cricketer cr:name ?name
```

Fig 2.1 code snippet

}

WHERE {

```
?cricketer rdf:type dbpedia-owl:Cricketer .
      ?cricketer foaf:name ?name .
```

Fig 2.2 code snippet

}

2. **Birthplace:** This class defines the birthplace of all the cricketers and its object property is 'hasBirth' with Domain(intersection) as Cricketer and Range(intersection) as Birthplace.

3. **Captain:** This class defines the captains for all over the formats for different teams/nations/leagues like Test, ODI, T-20, etc. Every team (nation wise) can have different captains for all the formats given above. Its object property is 'isCaptain' with Domain(intersection) as Captain.

4. **Mostruns:** This class defines the most runs scored by the cricketers in all over the formats. The data is stored for all the retired and non-retired cricketers. Its object property is 'hasMostRuns with Domain(intersection) as Mostruns.

5. **MostTestRuns:** This class defines the runs scored by the cricketers in Test format specifically, it says most runs in a Test match. This includes the records of both international and domestic matches. Its object property is 'hasMostTestRuns' with Domain(intersection) as MostTestRuns.

6. **Odiruns:** This class defines the cricketers scoring most ODI runs in an ODI format. This holds the records of both International and Domestic matches. Its object property is 'mostOdiRuns' with Domain(intersection) as Odiruns.

2.2. Asserted properties

In my case I am asserting 3 **Datatype properties**:

- **batting** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: string

- **bowling** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: string
- **clubnumber** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: integer
- **dateofbirth** = The Domains(intersection) of batting is Cricketer and its Range(intersection)is xsd: dateTime
- **deliveries** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: integer
- **height** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: double
- **lastodiagainst** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: string
- **lastodiyear** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: integer
- **name** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is xsd: string

And I am asserting the following **Object properties**:

- **hasBirth** = The Domains(intersection) of batting is Cricketer and its Range(intersection) is Birthplace
- **hasMoreTestRuns**= The Domains(intersection) of batting is Cricketer and its Range(intersection) is MostTestRuns
- **hasMostRuns**= The Domains(intersection) of batting is Cricketer and its Range(intersection) is Mostruns
- **isCaptain**= The Domains(intersection) of batting is Cricketer and its Range(intersection) is Captain
- **mostOdiRuns**= The Domains(intersection) of batting is Cricketer and its Range(intersection) is Odiruns
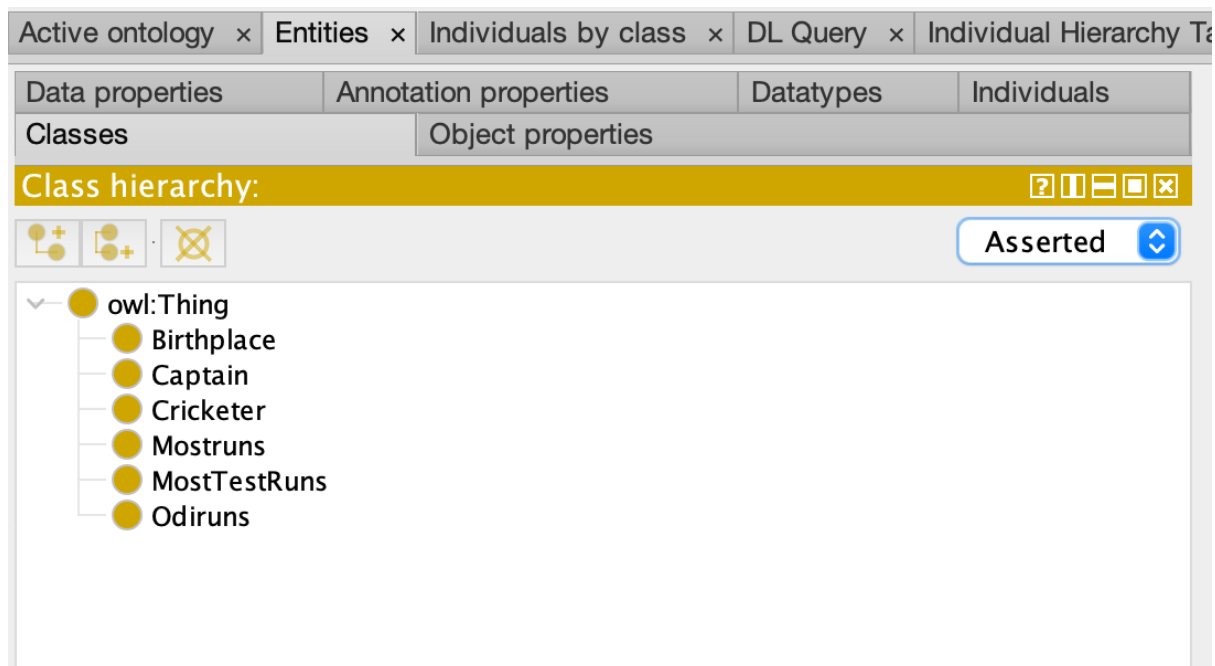
2.3. Class hierarchy

Fig 2.3 class hierarchy
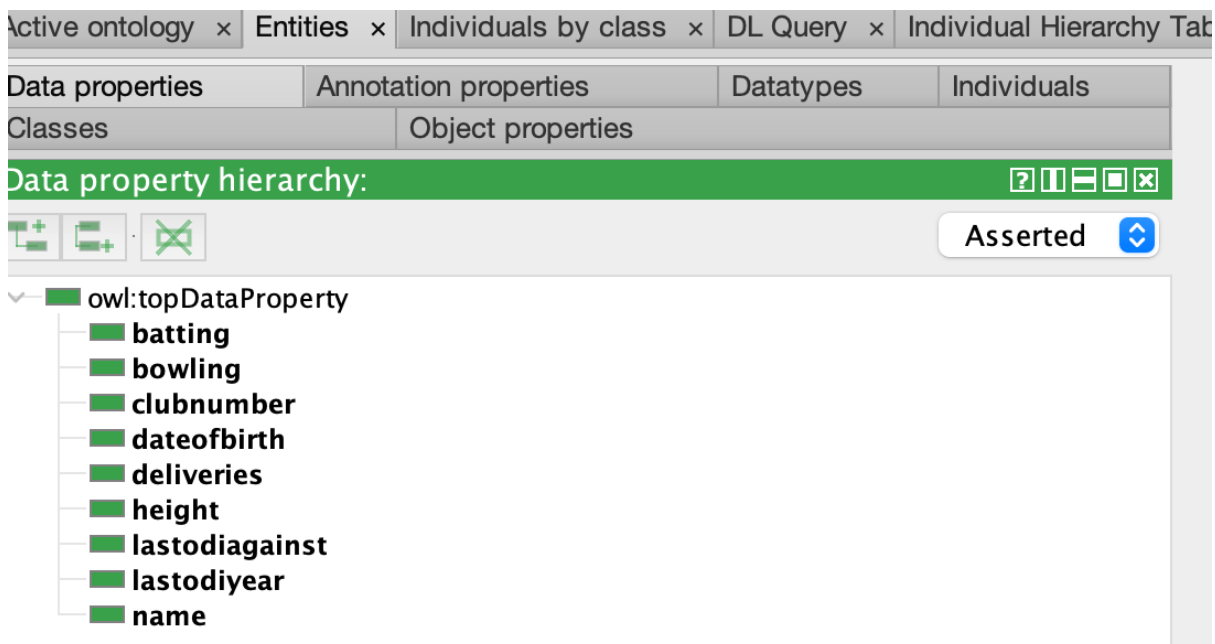
2.4 Data properties

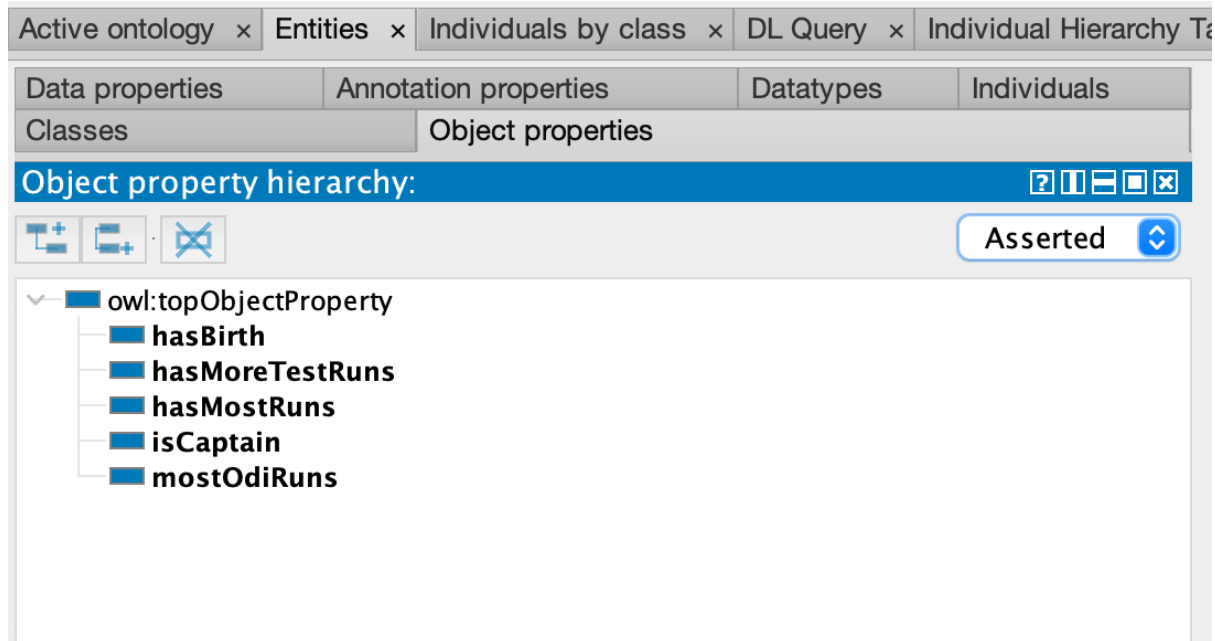*Fig 2.4 Data properties*

2.5 Object Properties

Fig 2.5 Object properties

3. Asserted individuals

We asserted the following individuals into our ontology.

Class	Individuals
Birthplace	Acton_Green,London, Anfield_(suburb), Ashford,_Kent, Ashford,_Surrey …
Captain	2007_Future_Cup, 2018–19_Oman_Quadrangular_Series, 2007_Future_Cup, Zimbabwean_cricket_team_in_Ireland_in_2021…

Cricketer	Jonathan_Hall_, Joseph_Needham,Mark_Chapman, Abdullah_Mazari …
Mostruns	Sir_Vivian_Richards_Trophy, Rajasthan_Royals_in_2011,Kolkata_Knight_Riders_in_2014, 2017–18_Sunfoil_3-Day_Cup …
MostTestRuns	English_cricket_team_in_India_in_1981–82, English_cricket_team_in_India_in_2008–09, English_cricket_team_in_India_in_2008–09 …
Odiruns	Indian_cricket_team_in_Pakistan_in_2003–04, Indian_cricket_team_in_England_in_1982, Bangladeshi_cricket_team_in_New_Zealand_in_2020–21 …

Table 3. Asserted individuals into ontology.

We discarded a few people from this table for brevity. For a full rundown of people, kindly allude to the philosophy record. Likewise attested a few proclamations to construe essential connections between these substances.

PREFIX cr: <http://www.semanticweb.org/uttkarshraj/ontologies/2022/4/proj_cricketer#>

```
CONSTRUCT {
      ?cricketer rdf:type cr:Cricketer .
      ?cricketer cr:dateofbirth ?dateofbirth .
      ?cricketer cr:name ?cricketplayers .
   }
   WHERE{
      ?cricketer wdt:P31 wd:Q5 .
      ?cricketer wdt:P106 wd:Q12299841 .
      ?cricketer rdfs:label ?cricketplayers .

      ?cricketer wdt:P27 wd:Q668 .
      ?cricketer wdt:P569 ?dateofbirth .
      FILTER(LANG(?cricketplayers) = "en") .
      SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
   }"""
```

Fig 3.1 code snippet

Note: We pulled the data from Wikidata and DBpedia and ran the query over them and later stored the values and joined two tables to compares its values.

```
SELECT DISTINCT ?cricketplayers ?dateofbirth
WHERE {
```

```
    ?cricketer rdf:type cr:Cricketer .
    OPTIONAL {?cricketer cr:name ?cricketplayers} .
    OPTIONAL {?cricketer cr:dateofbirth ?dateofbirth} .
    }GROUP BY ?dateofbirth"""
```

Fig 3.2 code snippet

The query above is giving me the values of all the names of the cricket players and their date of birth.

```
Dilip Ghose                                    1932-12-05T00:00:00+00:00

P. M. Raghavan                                 1920-12-18T00:00:00+00:00

Rahul Prasad                                   1982-01-03T00:00:00+00:00

Cheteshwar Pujara                              1988-01-25T00:00:00+00:00

Hemanga Baruah                                 1966-03-14T00:00:00+00:00

Rustomji Jamshedji                             1892-11-18T00:00:00+00:00

Rajani Venugopal                               1969-05-28T00:00:00+00:00

Ashay Sardesai                                 1998-01-09T00:00:00+00:00

P. M. K. Mohandas                              1948-01-31T00:00:00+00:00

Harsh Tyagi                                    1999-12-23T00:00:00+00:00

Manan Sharma                                   1991-03-19T00:00:00+00:00
```

Fig3.3 Cricket players and Date of Birth

4. Issues and resolutions

During the execution of the Cricketer ontology, we ran into the accompanying issues:

⇨ We additionally note that clients might experience out-of-memory issue while running reasoner. To fix that, possibly you really want to build the Java load space or decrease the arrangement of rules you need to utilize.

5. Conclusion

Fig5 Output of query_basic.py file

Fig5.1 Output of query_bonus.py file

6. Final Section

Note: There are total of 7 files (**4** '.py' files, **3** '.owl' files) + **1**'.pdf' file for report.

   i)   basic.py
   ii)   query_basic.py
   iii)   bonus.py
   iv)   query_bonus.py
   v)   proj_cricketer.owl
   vi)   cricketer_basic_new.owl
   vii)   cricketer_bonus.owl

Step1: First we will import the folder 'Semantic Project' to the directory.

Step 2: We will then run the following commands:

   ⇨ *'python basic.py'* to run the basic.py file
   ⇨ '*python query_basic.py*' to run the query_basic.py file
   ⇨ '*python bonus.py*' to run the bonus.py file
   ⇨ '*python query_bonus.py*' to run the query_bonus.py file

Step 3: Check the terminal for the output.

Metrics

Axiom	**361**
Logical axiom count	**341**
Declaration axioms count	**20**
Class count	**6**
Object property count	**5**
Data property count	**9**
Individual count	**140**
Annotation Property count	0

Class axioms

SubClassOf	0
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0

Object property axioms

SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrrefexiveObjectProperty	0
ObjectPropertyDomain	**5**
ObjectPropertyRange	**5**
SubPropertyChainOf	0

Fig5. Active Ontology (Ontology Metrics)

Data property axioms

SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	9
DataPropertyRange	9

Individual axioms

ClassAssertion	149
ObjectPropertyAssertion	139
DataPropertyAssertion	25
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0

Annotation axioms

AnnotationAssertion	0
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Fig5.1 Active Ontology (Ontology Metrics)

References:

1. **https://dbpedia.org/ontology/Cricketer**
2. **https://www.wikidata.org**