CS6770: Knowledge Representation and Reasoning Assignmnent 1

Topic: ALC Taxonomy Builder - Subsumption Hierarchy

Teammate 1: Siddesh Hegde **Roll number:** CS21M064
Teammate 2: Gourav Sharma **Roll number:** CS21M020

Introduction:

The goal of this assignnment is to write a program to build the taxonomy (subsumption hierarchy) of concepts defined in the KB.

A concept D subsumes a concept C if $C^I \sqsubseteq D^I$ in every interpretation I; we write that the KB entails $C \sqsubseteq D$. For this assignment, the goal is to find all the concepts which subsume any concept C, and do this for all the concepts in the KB.

Given two concepts C and D, checking whether C is subsumed by D is a decidable procedure. This procedure to arrange all concepts into a taxonomy.

Executing the program:

1. **Language Used:** Python

2. **Required Libraries:**

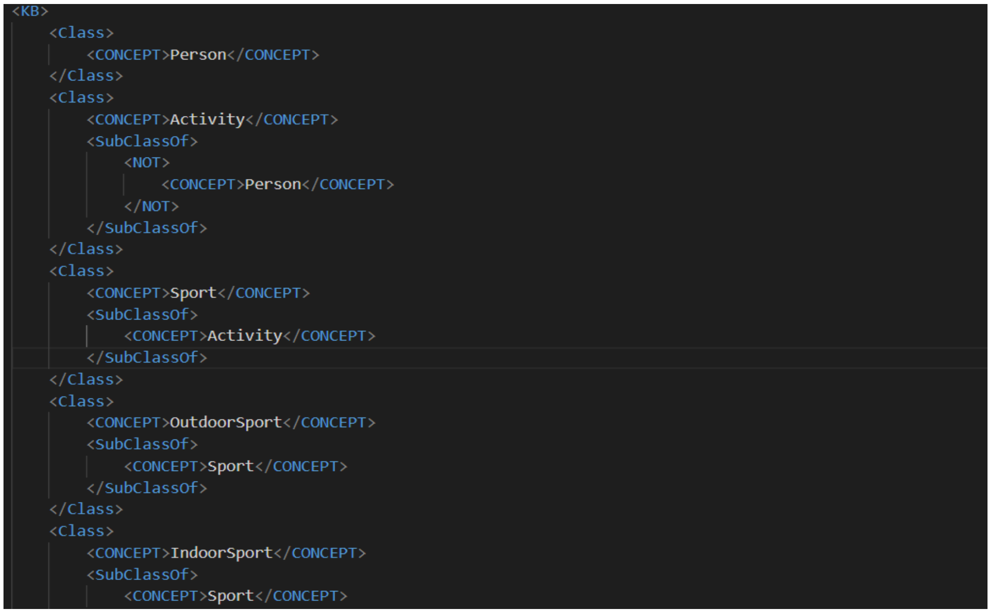
- xmltodict (pip install xmltodict)
- dict2xml (pip install dict2xml)
- pprint
- collections
- json

3. **File structure:**

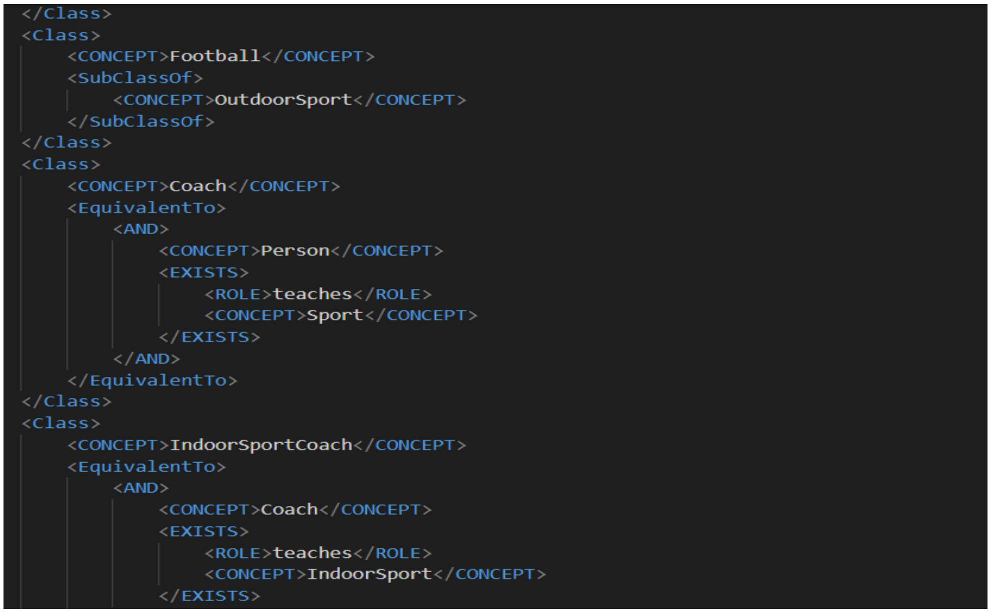
Keep the programs .py file and the input XML in the same directory. Once the code is executed, the output xml file is generated in the same directory.

4. **Input:**

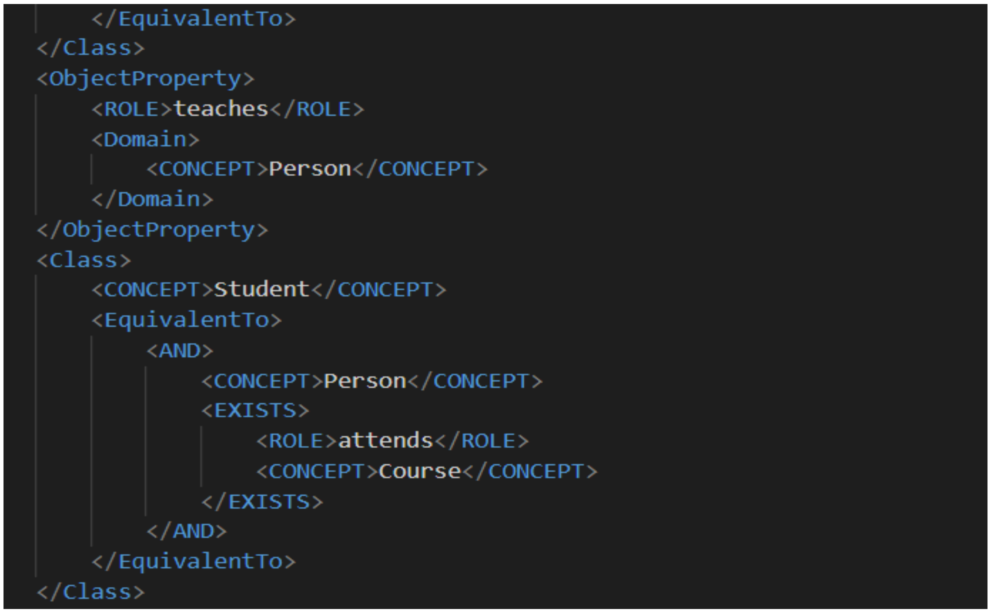
The input to the program is an XML file containing the tag structure representing the various class equivalence and class subsumption relations. The below image shows a sample input:



```
<KB>
    <Class>
        <CONCEPT>Person</CONCEPT>
    </Class>
    <Class>
        <CONCEPT>Activity</CONCEPT>
        <SubClassOf>
            <NOT>
                <CONCEPT>Person</CONCEPT>
            </NOT>
        </SubClassOf>
    </Class>
    <Class>
        <CONCEPT>Sport</CONCEPT>
        <SubClassOf>
            <CONCEPT>Activity</CONCEPT>
        </SubClassOf>
    </Class>
    <Class>
        <CONCEPT>OutdoorSport</CONCEPT>
        <SubClassOf>
            <CONCEPT>Sport</CONCEPT>
        </SubClassOf>
    </Class>
    <Class>
        <CONCEPT>IndoorSport</CONCEPT>
        <SubClassOf>
            <CONCEPT>Sport</CONCEPT>
```



```
    </Class>
    <Class>
        <CONCEPT>Football</CONCEPT>
        <SubClassOf>
            <CONCEPT>OutdoorSport</CONCEPT>
        </SubClassOf>
    </Class>
    <Class>
        <CONCEPT>Coach</CONCEPT>
        <EquivalentTo>
            <AND>
                <CONCEPT>Person</CONCEPT>
                <EXISTS>
                    <ROLE>teaches</ROLE>
                    <CONCEPT>Sport</CONCEPT>
                </EXISTS>
            </AND>
        </EquivalentTo>
    </Class>
    <Class>
        <CONCEPT>IndoorSportCoach</CONCEPT>
        <EquivalentTo>
            <AND>
                <CONCEPT>Coach</CONCEPT>
                <EXISTS>
                    <ROLE>teaches</ROLE>
                    <CONCEPT>IndoorSport</CONCEPT>
                </EXISTS>
```



```
    </EquivalentTo>
</Class>
<ObjectProperty>
    <ROLE>teaches</ROLE>
    <Domain>
        <CONCEPT>Person</CONCEPT>
    </Domain>
</ObjectProperty>
<Class>
    <CONCEPT>Student</CONCEPT>
    <EquivalentTo>
        <AND>
            <CONCEPT>Person</CONCEPT>
            <EXISTS>
                <ROLE>attends</ROLE>
                <CONCEPT>Course</CONCEPT>
            </EXISTS>
        </AND>
    </EquivalentTo>
</Class>
```

5. **Logic:**

- The program extracts data from the various tags of the input XML file.
- This data is saved as a dictionary (key-value pairs)
- The code works in 2 passes
- The first pass processes each class one at a time and finds all the subclass and equivalence relations for that class from the input. This information is stored in separate lists for each class.
- The second pass uses the created list of the subclasses and equivalence relations and stores generates new tags for the output in a dictionary in the mentioned format for the output data.
- After the subclasses are generated, the information is read one class at a time and the output text file is generated using the data.

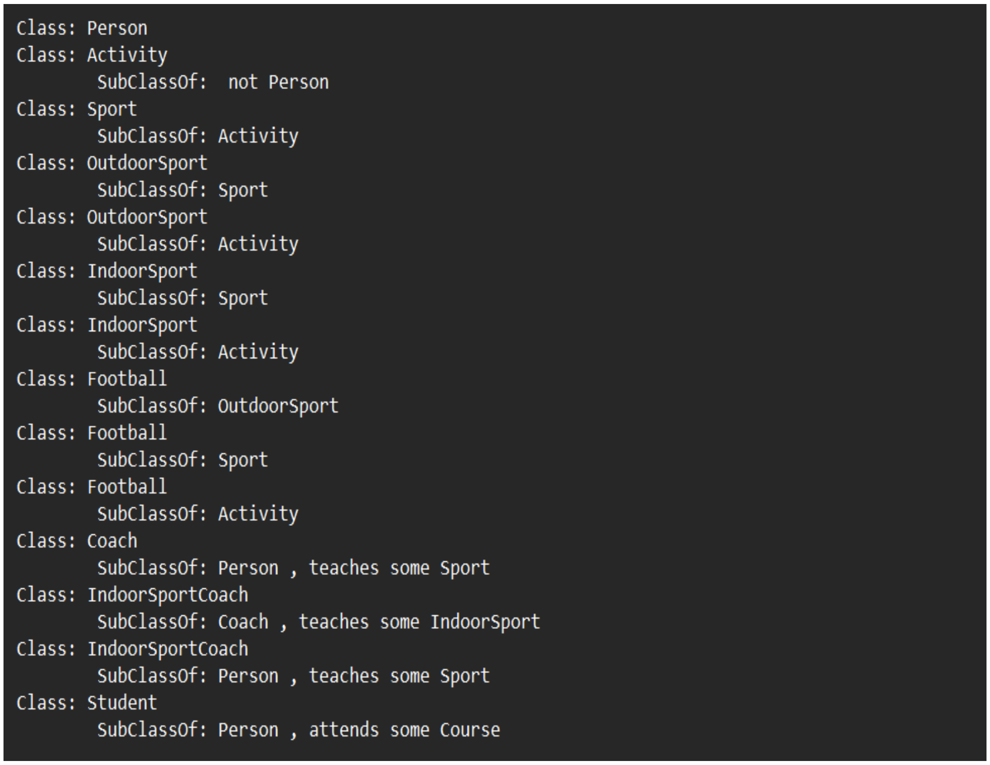
5. **Functions Used:**

- **save:** This functions takes the dictionary key-value pairs for output and generates the output text file

- **convertXMLToDict:** This function converts XML data into a dictionary (Key-Value pairs)
- **processTags:** This function takes in the input data in dictionary form and processes the data. The functions returns the output data as a dictionary
- **saveXMLOutput:** This function takes in the output data in dictionary form, converts to XML and returns it.

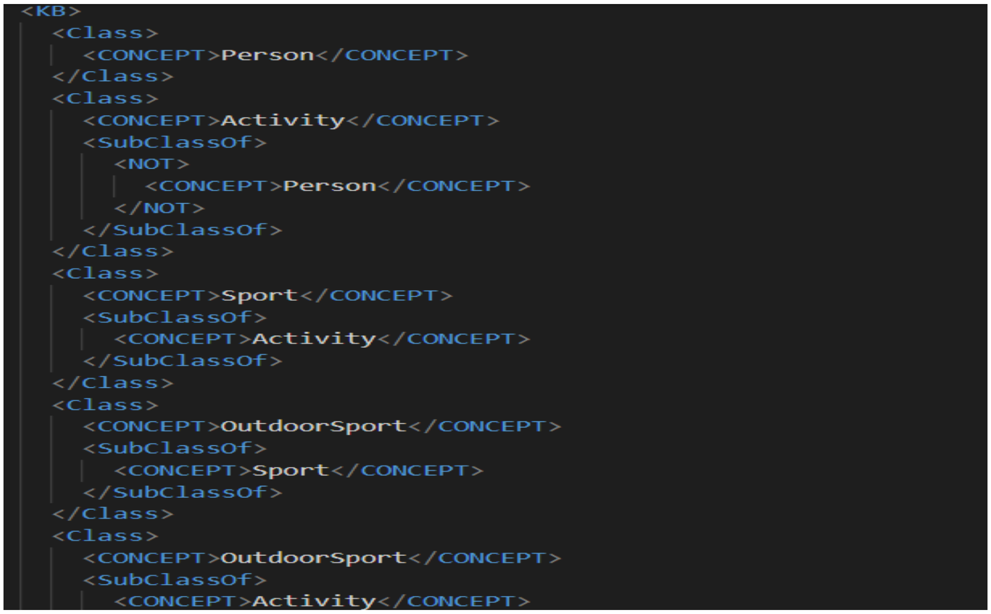
6. **Output:**

The output text file is generated which is stored in ”output.txt” file.

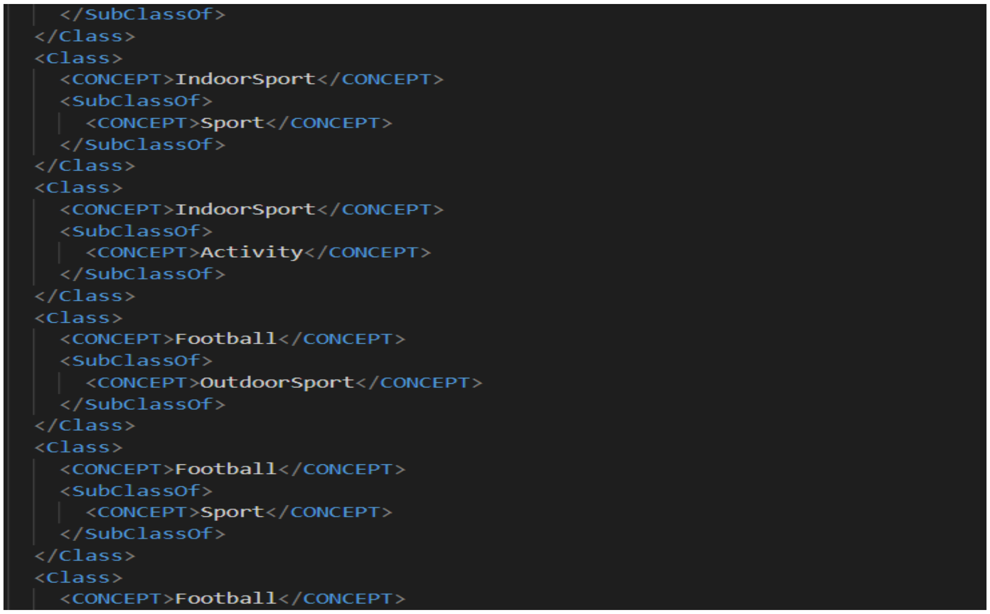


```
Class: Person
Class: Activity
        SubClassOf:  not Person
Class: Sport
        SubClassOf: Activity
Class: OutdoorSport
        SubClassOf: Sport
Class: OutdoorSport
        SubClassOf: Activity
Class: IndoorSport
        SubClassOf: Sport
Class: IndoorSport
        SubClassOf: Activity
Class: Football
        SubClassOf: OutdoorSport
Class: Football
        SubClassOf: Sport
Class: Football
        SubClassOf: Activity
Class: Coach
        SubClassOf: Person , teaches some Sport
Class: IndoorSportCoach
        SubClassOf: Coach , teaches some IndoorSport
Class: IndoorSportCoach
        SubClassOf: Person , teaches some Sport
Class: Student
        SubClassOf: Person , attends some Course
```

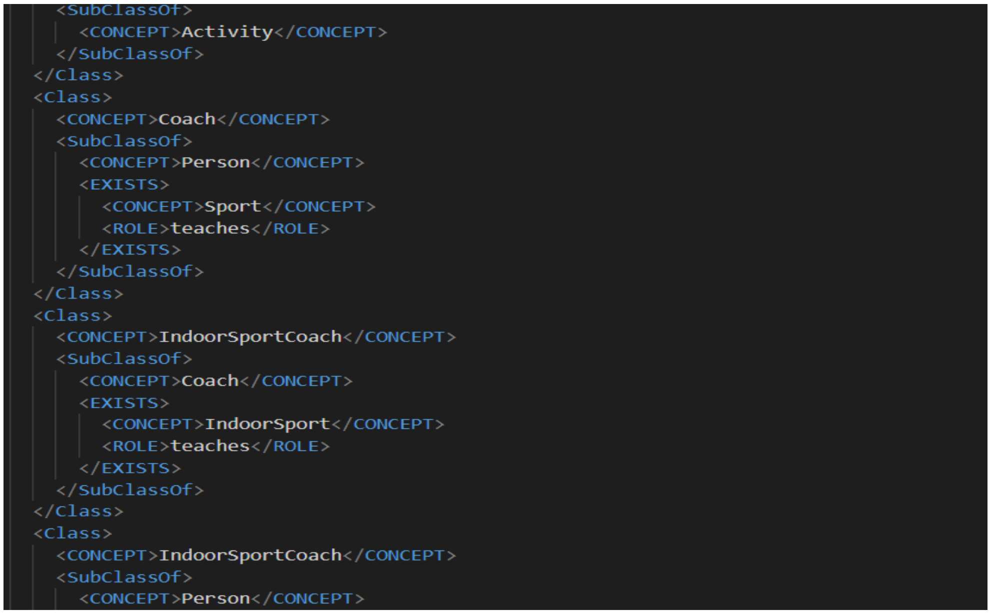
This output dictionary is the parsed so that the output XML can be generated from this. The final output generated is then saved as ”outputfile.xml”



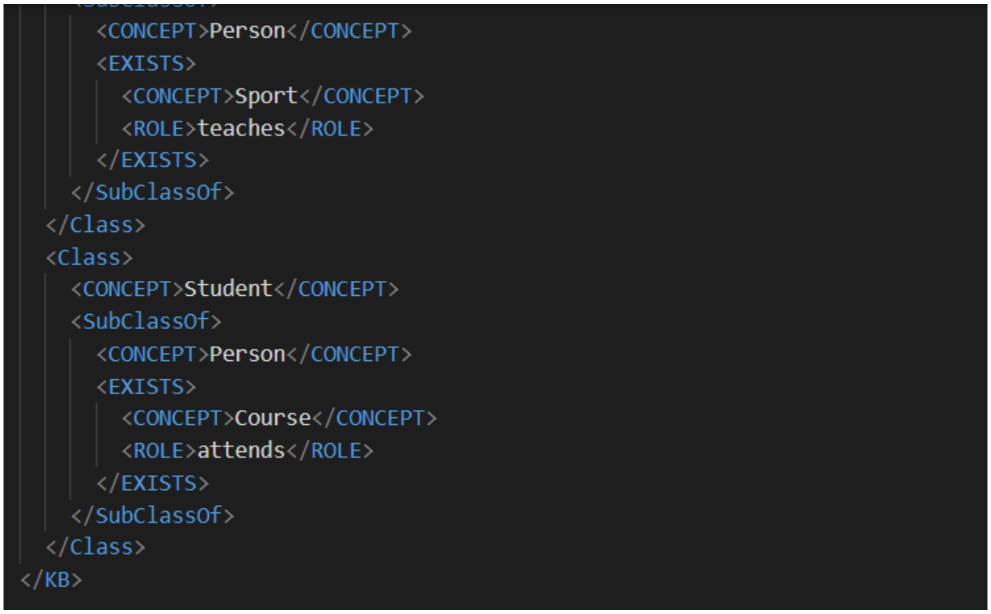
```
<KB>
  <Class>
    <CONCEPT>Person</CONCEPT>
  </Class>
  <Class>
    <CONCEPT>Activity</CONCEPT>
    <SubClassOf>
      <NOT>
        <CONCEPT>Person</CONCEPT>
      </NOT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>Sport</CONCEPT>
    <SubClassOf>
      <CONCEPT>Activity</CONCEPT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>OutdoorSport</CONCEPT>
    <SubClassOf>
      <CONCEPT>Sport</CONCEPT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>OutdoorSport</CONCEPT>
    <SubClassOf>
      <CONCEPT>Activity</CONCEPT>
```



```
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>IndoorSport</CONCEPT>
    <SubClassOf>
      <CONCEPT>Sport</CONCEPT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>IndoorSport</CONCEPT>
    <SubClassOf>
      <CONCEPT>Activity</CONCEPT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>Football</CONCEPT>
    <SubClassOf>
      <CONCEPT>OutdoorSport</CONCEPT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>Football</CONCEPT>
    <SubClassOf>
      <CONCEPT>Sport</CONCEPT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>Football</CONCEPT>
```



```
    <SubClassOf>
      <CONCEPT>Activity</CONCEPT>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>Coach</CONCEPT>
    <SubClassOf>
      <CONCEPT>Person</CONCEPT>
      <EXISTS>
        <CONCEPT>Sport</CONCEPT>
        <ROLE>teaches</ROLE>
      </EXISTS>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>IndoorSportCoach</CONCEPT>
    <SubClassOf>
      <CONCEPT>Coach</CONCEPT>
      <EXISTS>
        <CONCEPT>IndoorSport</CONCEPT>
        <ROLE>teaches</ROLE>
      </EXISTS>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>IndoorSportCoach</CONCEPT>
    <SubClassOf>
      <CONCEPT>Person</CONCEPT>
```



```
      <CONCEPT>Person</CONCEPT>
      <EXISTS>
        <CONCEPT>Sport</CONCEPT>
        <ROLE>teaches</ROLE>
      </EXISTS>
    </SubClassOf>
  </Class>
  <Class>
    <CONCEPT>Student</CONCEPT>
    <SubClassOf>
      <CONCEPT>Person</CONCEPT>
      <EXISTS>
        <CONCEPT>Course</CONCEPT>
        <ROLE>attends</ROLE>
      </EXISTS>
    </SubClassOf>
  </Class>
</KB>
```