Semantic Web (CSE632)
Winter 2020
Assignment 3

Instructions

1. Assignment is not a group activity. Each student has to work on it by himself/herself and submit the assignment.
2. Plagiarism check and corresponding policies will be strictly enforced. Students can be selected at random for a "viva" on the assignment and if the responses are not satisfactory, they will get a 0 in the assignment.
3. All the text and images should go into a single pdf file. All the code should be in a directory named "code" and the name of the files should be RollNo_Q#.java, where # is the question number. Include a jar file of the source code. OWL files should go into a directory named "ontologies" and name the owl file as RollNo_Q#.owl. Create a zip file with the name RollNo_HW3.zip and submit it.
4. Provide comments in the code and document every step of your code (except variable assignment). If the code is not documented well, there will be a penalty of 10% reduction in the points.

Questions **Max points: 50**

1. Consider the following TBox **5 pt**
   K = {Vegan≡Person⊓ ∀eats.Plant, Vegetarian≡Person⊓ ∀eats.(Plant⊔Dairy)}
   Check using ALC Tableau whether Vegan⊑Vegetarian is a logical consequence of K, or in other words K ⊢ Vegan⊑Vegetarian

2. Given the following concept names and role names,

   Concept names: Team, DebutTeam, ChampionshipTour, Tournament, SportGame, BasketballTeam, RugbyTeam, FootballTeam, VolleyballTeam, MultisportTeam, InterestingTournament, SmallSportEvent, Person, Trainer, Sportsman, TeamMember, StrongAthlete, Footballer, Master, TeamSport, PopularSport

   Role names and their characteristics
   a. consistsOf is transitive and its inverse is belongsTo
   b. hasWinner is a functional role, i.e., a tournament cannot have two different winners. Its inverse role is isWinnerOf.
   c. organizedBy, and its inverse is organizes
   d. plays, and its inverse is isPlayedBy
   e. includes, its inverse is isPlayedAt
   f. playsFor
   g. takesPartIn
   h. participatesIn

i. responsibleFor

i) Translate the following statements into description logics **15 pt**

- A team takes part in at least one championship tour.
- A championship tour is organized by some team, and consists of at least one tournament.
- A tournament belongs to some championship tour.
- A sportsman is someone who plays some sport game.
- A team member is a sportsman who plays for some team.
- A strong athlete is a sportsman who is a team member or plays at least three sport games.
- A footballer is someone who plays for a football team.
- A trainer is someone who trains (responsible for) some team.
- A master is someone who is both a sportsman and a trainer.
- A sport game is played at some tournament.
- A team sport is a sport game, which has at least one winner.
- A popular sport game is a sport game that is played at at least two tournaments
- A multisport team is a team that plays at least two of the known sport games (basketball + football, basketball + rugby, basketball + volleyball, football + rugby, football + volleyball or rugby + volleyball).
- A tournament is interesting, if its winner is a debut team.
- A small sport event is a tournament which includes no more than two sport games.

ii) Build an ontology (you can use Protege) that includes the above given concepts, role names, and the axioms. Make sure to define domain and range for every role. Write at least 10 competency questions for this ontology. **5 pt**

iii) Add the following ABox statements to the ontology created in the above step. **5 pt**

- BV is a basketball/volleyball team that participates in the two championship tours BVOct and BVNov, with the tournaments: BVOct1 and BVOct2, and BVNov1 respectively.
- Tournament BVOct1 is won by the BV team.
- B is a basketball team that participates in a single championship tour BOct with tournaments BOct1 and BOct2.
- Tournament BOct1 is won by the team B, which is a debut team.
- FR is a football/rugby team that takes part in a single championship tour FROct with the single tournament FROct1.
- Sportsman Bob plays SportGame1, which is played at tournaments BVOct1, BVOct2 and BVNov1.
- Sportsman Kate plays SportGame2, which is played at tournaments BVOct2 and BVNov1. The same sportsman also plays SportGame3, which is played at tournament FROct1, and sport game SportGame4, which is played at tournament BOct2.
- Sportsman Tim plays SportGame5, which is played at tournaments BOct1 and BOct2. Sportsman Tim also plays SportGame6, which is played at tournament FROct1.
- Person Mary is responsible for team BV.

- Sportsman Kate is responsible for team FR.

Using a reasoner (one of HermiT, Pellet, JFact or FaCT++) listed in Protege, check the consistency of the ontology. Submit a single owl file and the competency questions can go into the pdf file.

3. Model the data given in the Netflix file of Assignment-1 using an ontology. Make use of at least two ontology design patterns. For each class in your ontology, mention the three characteristics (rigidity, identity, unity) and justify your class hierarchy based on these characteristics. For each class, you should also mention if they are anti-rigid, anti-identity, or anti-unity. Add at least 10 TBox axioms in the ontology that are not subClassOf axioms and domain and range axioms for roles. Make sure to define domain and range for every role. Along with the owl file, submit the justification of the class hierarchy, and a diagrammatic representation of the classes and properties in the ontology. Use rectangles for classes and labelled edges for the properties between the classes (similar to the procedure followed in the ODP tutorials on Chess and Cooking Recipe papers uploaded on the Classroom). **10 pt**

4. Write a Java program using the OWL API and the OWL Explanation API (links given below) to connect with any two reasoners from among HermiT, Openllet, JFact, and do the following: for an ontology given as input, check its consistency, and perform classification. Write the classification output to a text file that has the following format

   Class-name (cn_i) = {comma separated classes that are superclasses of cn_i}
   Class-name (cn_j) = {comma separated classes that are superclasses of cn_j}

   Note that the class names should include just the local part and not the complete IRI. Compare the output of the two reasoners (use code to make this comparison) and write your comments.

   Use the OWL API and the Explanation API to generate explanations for the following
   - A (subClassOf) B, where A and B are either simple classes or complex classes (or class expressions) from the input ontology. A can also be an individual.
   - Given ontology is inconsistent and generate explanation as to why that is the case

   Comment on the difference in the usage or the output among the two APIs. **10 pt**

   OWL API: https://github.com/owlcs/owlapi
   OWL API wiki: https://github.com/owlcs/owlapi/wiki
   OWL Explanation: https://github.com/matthewhorridge/owlexplanation
   Openllet: https://github.com/Galigator/openllet

HermiT: http://www.hermit-reasoner.com/, https://github.com/phillord/hermit-reasoner
JFact: https://github.com/owlcs/jfact, http://jfact.sourceforge.net/

5. This is an optional question for bonus points. **15 pt**

An ontology is normalized to convert the axioms into a particular form (which is generally simpler compared to the original axioms). The normalization rules for an ontology in OWL 2 EL profile (description logic EL++) is given in the paper titled "Pushing the EL Envelope" (paper will be uploaded to the Classroom as well) on page 7. There are 7 normalization rules, numbered NF1 to NF7 (it will be a good idea to read until at least Section 3.1). The advantage of normalization is, all the axioms will be in a fixed form. The disadvantage is that the normalization process introduces several new classes and roles (which are redundant).

Example

$A \sqcap B \sqcap \exists r.C \sqsubseteq D$

Since this axiom does not satisfy any given normal form, it has to be normalized by applying NF2. On applying NF2, we get the following axioms

$A \sqcap E \sqsubseteq D$
$B \sqcap \exists r.C \sqsubseteq E$

Write a Java program (using OWL API) to denormalize a given normalized ontology, i.e., remove the redundant classes and roles. Each of the 7 rules now have to be applied in the reverse. Reduce the ontology to as small an ontology as possible.

In the above example, after denormalization, class E will be eliminated and we can replace the two axioms with the initial axiom ($A \sqcap B \sqcap \exists r.C \sqsubseteq D$) in the ontology.

Normalization code is available at
https://github.com/raghavam/DistEL/blob/master/src/knoelab/classification/init/Normalizer.java