CSI5180 Topics in AI - Ontologies and Semantic Web

Winter 2019

Version 2.0 - March 8th 2019

Assignment 3 - Reasoning and Ontology Matching

Objective	Become familiar with reasoning and ontology matching.
Due date	March 21, 11:30pm.
What to submit ?	Submit a report, in pdf format, which will include a title page (name, student number) as well as answers as specified for each of the questions below. Also submit the code you programmed for the matching.
How to submit ?	In Brightspace, through the link provided under the Assignment module.
Penalty	-10% per day late
Percentage	20% of overall semester grade
Software Requirement	You can program in Java/Python, as you wish.

PART 1 - DESCRIPTION LOGICS AND REASONING (30 points)

Question 1 - Description Logic and OWL (5 points)

We focused in class on the ALC (Attributive Language with Complements) flavor of Description Logics (DL). But in your Assignment 2, you were asked to explore all the possible concept and role constructs and restrictions allowed by OWL, which corresponds to a more expressive DL than ALC.

Please go back to the OWL ontology you built in Assignment 2:

IN YOUR REPORT:

a) provide two examples of concepts or roles which are within the ALC

b) provide two examples of concepts or roles which are NOT within the ALC (too expressive)

c) In Description Logics, we think of a Knowledge Base as being the combination of a TBox and a ABox. Can you give two examples from your ontology that would be part of the TBox, and two examples of instances you would put in the ABox.

Question 2 - Reasoning with Tableaux Algorithm (20 points)

Here is a small Knowledge Base, containing a T-Box and an A-Box.

Dietary KB

T-Box	MeatEater $\sqsubseteq$ $\forall eats.$(Chicken $\sqcup$ Beef) Vegetarian $\sqsubseteq$ $\forall eats.\neg$Meat Vegan $\equiv$ Vegetarian $\sqcap$ $\forall eats.$($\neg$RecipeWithDairy $\sqcap$ $\neg$Dairy) Fruitarian $\sqsubseteq$ $\forall eats.$ Fruit Strangetarian $\sqsubseteq$ $\forall eats.$(Fruit $\sqcup$ Beef) RecipeWithDairy $\equiv$ $\exists madeWith.$Dairy IceCream $\sqsubseteq$ Dairy Chicken $\sqsubseteq$ Meat Beef $\sqsubseteq$ Meat
A-Box	madeWith(sundae, chocolate-ice-cream) IceCream(chocolate-ice-cream) Fruit(tangerine)

We've defined a few typical reasoning tasks in class. Let's review each type of reasoning, and attempt to prove (or disprove) by refutation (using Tableaux Algorithm) each of the statements below.

	Type of Reasoning	Query
1	Concept subsumption	Q1- Fruitarian $\sqsubseteq$ Strangetarian Q2 - $\exists eats.$IceCream $\sqsubseteq$ $\neg$Vegan
2	Concept satisfiability: Is the concept empty (impossible)?	Q3 - Strangetarian $\equiv \bot$
3	Concept disjointness: Is Meateater disjoint from Vegetarian	Q4 - MeatEater $\sqcap$ Vegetarian $\equiv \bot$
4	Instance checking	Q5 - eats(john, sundae), Vegan(john) Q6 - eats(eric, tangerine), Fruitarian(eric) Q7 - eats(bo, chocolate-ice-cream), Fruitarian(bo)

IN YOUR REPORT:

Show your use of Tableaux algorithm to accept or refute the statements (Q1 to Q7) above. You can do the Tableaux Algorithm on paper, and just scan the papers to include in the pdf report.

You've probably arrived at the conclusion that one could be eating chocolate ice cream and still perhaps be considered a Fruitarian (Q7). Why is that? What is missing in the Diet KB? Please fix it so that we cannot arrive at that conclusion.

Question 3 - Investigating Reasoning tools (5 points)

Reasoning by hand, as in part 2, is not trivial... And people have programmed reasoners that can be used. I would like you to investigate what is available "out there" in terms of reasoning tools. Find at least 2 reasoners.

IN YOUR REPORT: For each of the two reasoners found, describe the reasoner, give a link to it, and mention which flavor of Description Logics it is able to handle.

PART 2 - ONTOLOGY MATCHING (30 points)

Ontology Matching can be performed by gathering evidence from different aspects (labels, structure, instance, model). Combining the evidence can then be done in various ways (sequential, parallel, iterative-approach to optimization).

Let's consider the two small food related KBs below. They do not contain roles, only concepts and instances, as well as a subsumption structure. The TBox is therefore a simple taxonomy for both KBs.

Knowledge Base	Classes	Instances	Subsumption
Food1	FruitOrVeggie Fruit StoneFruit CitrusFruit Veggie LeafyGreen Plant-Based Edible	Veggie(broccoli) LeaffyGreen(kale) StoneFruit(peach) CitrusFruit(tangerine) CitrusFruit(orange) Fruit(strawberry) Fruit(apple) Plant-Based(soymilk) Plant-Based(oatmilk)	Fruit $\sqsubseteq$ FruitOrVeggie Veggie $\sqsubseteq$ FruitOrVeggie StoneFruit $\sqsubseteq$ Fruit CitrusFruit $\sqsubseteq$Fruit LeafyGreen$\sqsubseteq$ Veggy Plant-Based $\sqsubseteq$ Edible FruitOrVeggie $\sqsubseteq$ Edible
Food2	Dairy IceCream Plant Fruit RedFruit Vegetable Food	RedFruit(strawberry) RedFruit(raspberry) IceCream(strawberry) Vegetable(broccoli) Vegetable(lettuce) Dairy(milk) Fruit(tangelo) Fruit(orange) Plant(oat)	RedFruit $\sqsubseteq$ Fruit Fruit $\sqsubseteq$ Plant Vegetable $\sqsubseteq$ Plant Plant $\sqsubseteq$ Food Dairy $\sqsubseteq$ Food IceCream $\sqsubseteq$ Dairy

a) Class labels comparison (6 points)

Program (in Java or Python) any approach seen in class for label comparison, calculate the similarity/distance between all pairs of classes, in Food1 and Food2. Identify some matches. Explore being very strict or very loose in your matching criteria and explore the results.

b) Class instances comparison (6 points)

You can reuse the approximate-string-matching approach from (a) to compare instances. And then program a class similarity based on shared instances (using Jaccard for example).

c) Taxonomy comparison (6 points)

Using any approach seen in class (such as upward co-topic distance, children comparison, bounded patch matching), program a class similarity based on taxonomy.

d) Merging strategy (6 points)

Implement a simple strategy (e.g. parallel merge) in which you gather the results from (a), (b), (c).

e) Discussion (6 points)

Discuss the results obtained. Which evidence (labels, instances, structure) seems more reliable? What is more problematic?

IN YOUR REPORT: For each of the questions (a) to (d), provide the snippet of code + results. For (e), just write a discussion on the results obtained.