Assignment #2 of Semantic Web

Ali Reza Barkhordari
Aban 3, 1394

Deadline: Aban 14, 1394 Dr. Morteza Amini

Principles

Students are supposed to follow the following rules.

• Once you are done, pack all your documents (if there are multiple ones) into a single ZIP file and send to abarkhordari@ce.sharif.edu. Please format your E-Mail’s title like the item below.

  SW - [Your Student ID] - [Assignment Number]
  example: SW-94111111-2

• Deadlines are unlikely to get extended, hence it is highly recommended to do your assignments by their deadlines.

• If you submit your assignments late, the following rules apply:
  – Up to 1 day late, you will lose 25% credits of the assignment at hand.
  – Up to 2 days late, you will lose 50% credits of the assignment at hand.
  – Up to 3 days late, you will lose 75% credits of the assignment at hand.

• You can turn in your assignments in either hand-written or typed format. However typed formats would be more welcomed. Nevertheless, if you would prefer hand-written format, you are expected to:
  – Write legibly!
  – Scan your documents and prepare a well-known image type (e.g. jpg) for sending by E-Mail.

• There will be a zero-tolerance policy for cheating/plagiarism.

• For practical exercises, You are welcome to use any techniques you would want to use.

• Kindly drop me an E-Mail, if you need further information or somethings seem unclear.
1 Theoretical Part

Problem 1.

Suppose we have the following model in Description Logic (DL).

\[
\Delta = \{ \text{Style}, \text{BlankSpace}, \text{IKnowPlaces}, \text{SlowAcid}, \text{Sirens}, \text{Taylor}, \text{Ellie}, \text{Calvin}, \text{Cher} \}
\]

\[
\mathcal{I}(\text{Song}) = \{ \text{Style}, \text{BlankSpace}, \text{IKnowPlaces}, \text{SlowAcid}, \text{Sirens} \}
\]

\[
\mathcal{I}(\text{Singer}) = \{ \text{Taylor}, \text{Ellie}, \text{Calvin}, \text{Cher} \}
\]

\[
\mathcal{I}(\text{Sings}) = \{ < \text{Taylor}, \text{Style} >, < \text{Taylor}, \text{BlankSpace} >, < \text{Taylor}, \text{IKnowPlaces} >, < \text{Calvin}, \text{SlowAcid} >, < \text{Cher}, \text{Sirens} > \}
\]

Table 1: The model

1. List all instances of the following statements in this model.
   - \( \forall \text{Sings}. \{ \text{Taylor} \} \)
     
     Answer
     
     The answer is \( \Delta - \{ \text{SlowAcid}, \text{Sirens} \} \).
     
     \[
     \{ \text{Style}, \text{BlankSpace}, \text{IKnowPlaces}, \text{Taylor}, \text{Ellie}, \text{Calvin}, \text{Cher} \}
     \]
   
   - \( \leq_2 \text{Sings} \)
     
     Answer
     
     The answer is \( \Delta - \{ \text{Taylor} \} \).
     
     \[
     \{ \text{Style}, \text{BlankSpace}, \text{IKnowPlaces}, \text{SlowAcid}, \text{Sirens}, \text{Ellie}, \text{Calvin}, \text{Cher} \}
     \]
   
   - \( \exists^{\leq 3} \text{Sings}. (\text{Singer} - \exists \text{Signs}. \text{Song}) \)
     
     Answer
     
     The answer is \( \Delta \).

2. What does the following statement mean in natural language? (Don’t translate it word by word, you’ve instead to do it in the best sensible way)

\[
\text{Sings} : \text{Taylor} \sqcap \exists \text{Sings}. \{ \text{Calvin} \}
\]

Answer

Songs sung by both Taylor and Calvin.
3. Write DL statements for the following natural language statements.

- Songs that have not been sung by any singers.

  Answer

  \[ \text{Song} \rightarrow \exists \text{Sings} . \text{Singer} \]

- Songs that have been sung by more than one singer.

  Answer

  \[ \exists > 1 \text{Sings} . \text{Singer} \]

- Singers who have not yet sung any songs.

  Answer

  \[ \text{Singer} \rightarrow \exists \text{Sings} . \text{Song} \]

4. Consider our model "as is" and also initially assume we’ve not Unique Name Assumption (UNA) applied on our model. Under these circumstances, if we were to add \((\leq 1 \text{Sings})(\text{Taylor})\) to A-Box of this model, then, is it satisfiable? How about if we assume UNA? (Please justify your answer)

  Answer

  In the former case, the answer is yes. However, in the latter case, it’s no. The justification is straightforward enough.

---

Problem 2.

Translate the following Description Logic (DL) statement to it’s corresponding/equivalent First-Order Logic (FOL) statement.

\[ \exists \geq 2 R_1 . (\forall R_2 . (\neg C_1 \sqcup C_2)) \]

**Hint:** You should start from \((\exists \geq 2 R_1 . (\forall R_2 . (\neg C_1 \sqcup C_2)))^\pi(x)\) and break it down step by step using applicable rules provided in the course until you can’t do anymore.

**Note:** Your answer should include all your taken steps. Final answer on its own will not be acceptable.

**Answer**

The answer is straightforward enough! Because of being too lengthy, I’ll omit.
Problem 3.

Which one is a tautology? Prove your answer by Tableaux algorithm.

- $(\neg A \lor B) \leftrightarrow (B \rightarrow A)$

<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>As this statement occasionally evaluates to false for a certain combination of $A$ and $B$, this is not, therefore, a tautology.</td>
</tr>
</tbody>
</table>

- $(\forall R.(\forall S.A)) \cap (\exists R.(\forall S.B)) \cap (\forall R.(\exists S.C)) \sqsubseteq \exists R.(\exists S.(A \cap B \cap C))$

- Note that in this formula $R$ and $S$ are of type roles and $A$, $B$ and $C$ are of type concepts.

<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is a tautology.</td>
</tr>
</tbody>
</table>

**Proof:**

For proving this statement, it’s enough to prove the following concept is not satisfiable.

$(\forall R.(\forall S.A)) \cap (\exists R.(\forall S.B)) \cap (\forall R.(\exists S.C)) \cap \neg \exists R.(\exists S.(\neg A \cup \neg B \cup \neg C))$

This predicate in Negative Normal Form (NNF) is as follows.

$(\forall R.(\forall S.A)) \cap (\exists R.(\forall S.B)) \cap (\forall R.(\exists S.C)) \cap \forall R.(\exists S.(\neg A \cup \neg B \cup \neg C))$

Tableaux algorithm.

$A_0 = \{(\forall R.(\forall S.A)) \cap (\exists R.(\forall S.B)) \cap (\forall R.(\exists S.C)) \cap \forall R.(\exists S.(\neg A \cup \neg B \cup \neg C))(x)\}$

$A_1 = A_0 \cup \{(\forall R.\forall S.A)(x), (\exists R.\forall S.B)(x), (\forall R.\exists S.C)(x), (\forall R.\forall S.(\neg A \cup \neg B \cup \neg C))(x)\}$

From $(\exists R.\forall S.B)(x)$, we can obtain:

$A_2 = A_1 \cup \{R(x, y), (\forall S.B)(y)\}$

Using Role Value Restriction rule, we have:

$A_3 = A_2 \cup \{(\forall S.A)(y), (\exists S.C)(y), (\forall S.(\neg A \cup \neg B \cup \neg C))(y)\}$

Using Role Exists Restriction rule, we have:

$A_4 = A_3 \cup \{S(y, z), C(z)\}$

Using Role Value Restriction, we have:

$A_5 = A_4 \cup \{A(z), B(z), (\neg A \cup \neg B \cup \neg C)(z)\}$

At this point, the generated tree by the algorithm is divided into three branches, and each of which will clash.

- If $A_6 = A_5 \cup \{\neg A(z)\}$ then clashes with $\{(A(z))\}$.
- If $A_6 = A_5 \cup \{\neg B(z)\}$ then clashes with $\{B(z)\}$.
- If $A_6 = A_5 \cup \{\neg C(z)\}$ then clashes with $\{C(z)\}$ (from step $A_4$).

Therefore, that NNF statement is not satisfiable and thereby in original statement, the right side always subsumes the left side.

Sorry for being literally overkill. :(
Problem 4.

Compare Knowledge-bases and Databases. What is the differences between this two concepts? Your comparison should include the following critical edges.

- **Open World Assumption**
  
<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge-bases are subjected to Open World Assumption, while databases are subjected to Closed World Assumption.(^a)</td>
</tr>
</tbody>
</table>

- **Dynamic/Static nature of contents**
  
<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge-bases are of dynamic nature, while databases are of static nature.(^a)</td>
</tr>
<tr>
<td>^aAttribution: <a href="http://www.slideshare.net/mcjenkins/knwoedgebase-vs-database">http://www.slideshare.net/mcjenkins/knwoedgebase-vs-database</a></td>
</tr>
</tbody>
</table>

- **Inferencibility**
  
<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge-bases are inferencible, whereas databases are not.</td>
</tr>
</tbody>
</table>
2 Practical Part

Problem 1.

In this part of our assignment we’re going to get ourself started with Protégé tool. It’s homepage describes itself as “A free, open-source ontology editor and framework for building intelligent systems”. You may forthwith after reading this sentence ask yourself what an ontology and/or an intelligent system is? If so, don’t panic, you’ll get familiar with later in the course.

What we’ve aimed for this assignment could be enumerated as follows. (Please note that these’re not our deliverable items, they just lay out our assignment’s boundaries.)

1. Setting things up: Please install it on your machine and play around with to see how it looks like. You can download it at http://protege.stanford.edu/

2. Discovery: What’s it? What’s it used for? What features does it provide? In short, how awesome it is?

3. Modeling: We’re to model a tiny and simplified version of family relationships there and see how could we utilize our knowledge about Description Logic (DL) over there.

You can follow up a good tutorial from this link.

A Good Tutorial on Protégé

Sound interesting? I definitely think so! Okay, let’s get our hands dirty right away. Start off your experience by writing down in half of a paper how it was, Your write-up should include the following items.

• Enumerate a couple of requirements for which Protégé can be used. (Explain each one as much as you think is needed.)

<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>The answer could be too broad. For example:</td>
</tr>
<tr>
<td>– As an inference engine for existing knowledges.</td>
</tr>
<tr>
<td>– Association with other semantic technologies.</td>
</tr>
<tr>
<td>– Creating and organizing knowledges and ontologies.</td>
</tr>
</tbody>
</table>

• What facilities does it provide for Description Logic (DL)?

<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>– Description Logic (DL) query.</td>
</tr>
<tr>
<td>– Providing reasoners (HermitT, SAT Solver).</td>
</tr>
<tr>
<td>– Adding query as a fact to ontology.</td>
</tr>
</tbody>
</table>

• Regarding your experiences so far and what you’ve studied, can you determine whether Unique Name Assumption (UNA) is enabled by default in Protégé or not.

<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. By default Protégé assumes all individuals point to a single individual in real world.</td>
</tr>
</tbody>
</table>
In next part of this problem we’d want to model a simplified version of family relationships in Protégé.

1. Create a model in Protégé from scratch which yields the graph above. (Note that blue circles are classes and gray ones are individuals)

   **Answer**
   
   Just created! who cares? :)

2. Write these statements in DL Query and determine which individuals they would include.
   
   - Those who have some sons.
     
     **Answer**
     
     Person and HasChild some Man
   
   - Those who have less than two children.
     
     **Answer**
     
     Person and HasChild max 1
   
   - Those men who have only daughters.
     
     **Answer**
     
     Man and HasChild only Woman

3. Did you faced a weird behavior\(^1\) while you’re getting instances from the last two statements in previous question? If you didn’t, you rock! If you did, can you reason it out?

   **Answer**
   
   Yes, I did. That’s due to Open World Assumption being enabled in Protégé. Regarding this, therefore, we can’t claim a given man only has daughter. He could have a son, but we don’t know.

---

\(^1\)You may ask me what do you mean by “weird behavior”? I do mean that a situation in which you’re not getting the outcome which is the most expected one.
Deliverable Items:

- A half-page fitted write-up which addresses three items mentioned previously.
- Your model’s file exported in XML/OWL format.
- A written report in which you’ve answered other questions.

Problem 2.

This problem is optional. You’ll get extra marks by getting this done, though. This problem is also all about Protégé and is stated as follows.

Provided we’ve the model described in first question of theoretical part in mind, we’d want to extend it by several concepts listed below. First off, you’ll have to define that model in Protégé. By the time your modeling is going to be almost done, you could go ahead and consider adding the items below.

- Add three subclasses Rock, Pop and Country, which are all disjoint against each other, to the class Song.
- Add class Composer to the model. (Composers are those who are writing lyrics for songs.)
- Add relation Composes.

Write these statements in DL Query format. (Note that there’s no need to list their matching individuals here. The query on its own is enough.)

1. Songs that have been composed but never sung.

   Answer
   
   Song and not (inverse(Sings) some Singer) and (inverse(Composes) some Composer)

2. Composers who only composed country songs sung by Taylor.

   Answer
   
   Composer and Composes only (Country and inverse(Sings) value Taylor)

Note: You don’t need to send me your model’s files. Please note down your answers to these questions in your final written report. That would be enough.

Good Luck

Ali Reza Barkhordari
abarkhordari@ce.sharif.edu