Assignment #1 of Semantic Web

Ali Reza Barkhordari

Mehr 15, 1394

Deadline: Mehr 25, 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-94111111-1

• 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.

Please answer these questions about First-Order Logic (FOL).

1. If we have relation Takes(x, y) such that x is a student and y is a course, how could we tell that ”Ali takes either OS or ComputerSecurity (not both)”

   Answer
   \[
   \text{Takes}(\text{Ali}, \text{OS}) \leftrightarrow \neg \text{Takes}(\text{Ali}, \text{ComputerSecurity})
   \]

2. If we have relations Student(x) and Failed(x, y) such that x is a student and y is of type course, then can we tell ”There’s only one student failed course OS”? If so, write it’s FOL representation.

   Answer
   Yes.
   \[
   \exists x (\text{Student}(x) \land \text{Failed}(x, \text{OS}) \land \forall y (\text{Student}(y) \land \text{Failed}(y, \text{OS}) \rightarrow x = y))
   \]

3. One writes the following FOL statement for ”There is a student who is loved by every other students”

   \[
   \exists x (\text{Student}(x) \land \forall y (\text{Student}(y) \land \lnot (x = y) \land \text{Loves}(y, x)))
   \]

   Is it correct or not? If yes, why? If no, why?

   Answer
   No. This statement implies every thing in our model is a student which is obviously nonsense. Correct form follows.

   \[
   \exists x (\text{Student}(x) \land \forall y (\text{Student}(y) \land \lnot (x = y) \rightarrow \text{Loves}(y, x)))
   \]

4. In family relationships, how do you define great-grandfather (father of your grandfather)? Complete the following FOL statement. (You are only allowed to use relation Parent(x, y) where both x and y are of type persons.)

   \[
   \forall a, b \ \text{IsGreatGrandFatherOf}(a, b) \rightarrow ...
   \]

   Answer
   \[
   \forall a, b \ \text{IsGreatGrandFatherOf}(a, b) \rightarrow \exists c, d (\text{Parent}(a, c) \land \text{Parent}(c, d) \land \text{Parent}(d, b))
   \]

5. We would want to formulate graphs in FOL. Hence, we could define relation HasEdgeTo(x, y) where x and y are both of type graph nodes. As such, Can you write the following statements in FOL representation? If you can, do. If you cannot, why? (You are, of course, allowed to use relation Node(x) where x is a graph node.)
• A full graph. (A full graph is such a graph that we cannot add any extra edges to it)

Answer
\[ \forall x, y (\text{Node}(x) \land \text{Node}(y) \rightarrow \text{HasEdgeTo}(x, y)) \]

• An undirected graph. (A graph in which direction of edges does not matter)

Answer
\[ \forall x, y (\text{Node}(x) \land \text{Node}(y) \land \text{HasEdge}(x, y) \rightarrow \text{HasEdge}(y, x)) \]

• A Tree!

Answer
Trees cannot be modeled in FOL. The why would be we are unable to check cycles using FOL as trees are cycle-free.

Problem 2.

Search the web and list 4 applications of Semantic Technology in real world use cases.

For example we could add weather information or traffic data to road maps so that smart automobiles are able to decide which road is the best choice. In this example, actually we have structured the road maps’ data model. A well-structured data model could be integrated with other models as well. So, as you have already seen in this example we have simply integrated weather model to our road map model.

If you’d like to have a closer look at this example, we actually agreed an abstraction layer above our road maps’ data and weather data. And by which we have linked them together. This abstraction layer could be thought as an semantic agreement which our road maps’ model and weather’s model must abide by.

Answer
Many applications can be found in the following web pages:

• 5 things to know about the Semantic Technologies
• Semantic Web Case Studies and Use Cases

TL;DR
In short, applications can be listed as follows.

• A prominent use case would be in search engines. e.g. Google.
• Social networks e.g. Twitter, Facebook.
• StumbleUpon service.
• BBC broadcasting network.
2 Practical Part

Problem 1.

In this part of our assignment, we would like to get familiar with Alloy Analyzer and do a tiny project. As Wikipedia\(^1\) says ”Alloy provides a simple structural modeling tool based on first-order logic”. Thus, we could model our problems in Alloy using a FOL-liked language.

In this project we wouldn’t immerse ourself into deep concepts of Alloy and its language. However you are welcome to do so by your own as this is beyond of our project’s boundaries. For getting started, you could head the following link and start reading.

Welcome to A Guide to Alloy!

The Model

We would want to model a simplified version of education systems. Such that in which we have got only instructors, students and courses as well as two relations: taking a course by students and teaching them by lecturers. Also we should consider prerequisites for courses. Therefore, we could express them in Alloy language as follows.

```alloy
module our_tiny_model

sig Course {
    preq: set Course
}

sig Instructor {
    teaches: set Course
}

sig Student {
    takes: set Course
}

Okay, let’s complete this model. The rest is up to you! Requirements are listed below. Please add them as facts to our model.

- Prerequisite chain should not have cycles or self loops.

  Answer

  ```alloy
  fact {
      all c:Course | c not in c.^preq
  }
  ```

- Students cannot take courses which their prerequisites already have not been taken.

  Answer

  ```alloy
  fact {
      all s:Student | s.takes.*preq in s.takes
  }
  ```

\(^1\)https://en.wikipedia.org/wiki/Alloy_(specification_language)
**Note 1:** Please be informed that you do not need to keep history of taken courses by students. Consider we have only one semester and this model is applying to.

**Note 2:** Keep our model as simple as possible.

For even more completeness, you are assigned to write the following predicates as `pred` statements in our model.

- There is an instructor who teaches *all* courses.

  **Answer**

  ```
  pred P1 {
    some i:Instructor | i.teaches = Course
  }
  ```

- There is a course which is prerequisite for *all* other courses.

  **Answer**

  ```
  pred P2 {
    some c:Course | c in (Course - c).preq
  }
  ```

- Each course is taught by *only one* instructor.

  **Answer**

  ```
  pred P3 {
    all c:Course | #(c.~teaches) = 1
  }
  ```

**Important Note:**

You might already be noticed that Alloy Analyzer, for generating instances, spawn as few as elements it can. So, for testing your model you may want to have more elements, you can enforce the Alloy Analyzer by writing a `fact` which tells Alloy Analyzer how many elements each signature should have.

```alloy
fact {
  #Course = 4
  and
  #Instructor = 2
  and
  #Student = 2
}
```

This `fact` makes Alloy Analyzer to set cardinalities 4, 2 and 2 for signatures Course, Instructor and Student respectively.

**What you should hand over for this problem:**

1. A tidy and well-documented `.als` file containing the whole model including two facts and three predicates.

2. Once you are done, get an instance from Alloy Analyzer and capture its diagram as an image file. You should send me this image.
You’ve been struggling with no luck? Send me E-Mails for any inquiries.

Good Luck
Ali Reza Barkhordari
abarkhordari@ce.sharif.edu