Artificial Intelligence

Assignment 2

Matthias Horbach

mhorbach@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: 09.05.2020, 23:59
Tutorial on: 10.05.2020

Group Name: Turing

Abhinav Ralhan (221202684), abhinavr8@uni-koblenz.de
Hammad Ahmed (221202832), hammadahmed@uni-koblenz.de
Muhammad Asad Chaudhry (221202659), maac@uni-koblenz.de
Vishal Vidhani (221202681), vvidhani@uni-koblenz.de

1 Modeling in first-order logic (20 Points)

Formalize the statements below in first-order logic. Use only the following predicate symbols and equality for this:

- professor (of arity 1)
- confused (of arity 1)
- studentOf (of arity 2)

The statements are:

1. Peter is Alice's student. Alice is also someone's student.
2. Every person with a student is a professor.
3. Every professor is confused.
4. Anyone who is not a confused cannot be a professor.
5. No confused person is both a professor and anyone's student.
6. There is a confused person who is not a professor.
7. Peter and Mary are students of the same professor.
8. People who are confused are not students of a non-confused professor.
9. A student is confused if, and only if, they have a confused professor.
10. Peter has at least two confused professors.

Solution:

1. $\text{studentOf}(\text{Peter}, \text{Alice}) \wedge \exists x(\text{studentOf}(\text{Alice}, x))$
2. $\forall x \forall y(\text{studentOf}(x, y) \rightarrow \text{professor}(x))$
3. $\forall x(\text{professor}(x) \rightarrow \text{confused}(x))$
4. $\forall x(\neg\text{confused}(x) \rightarrow \neg\text{professor}(x))$
5. $\forall x(\text{confused}(x) \rightarrow \neg(\text{professor}(x) \wedge \exists y(\text{studentOf}(y, x))))$
6. $\exists x(\text{confused}(x) \wedge \neg\text{professor}(x))$
7. $\exists p(\text{studentOf}(\text{Peter}, p) \wedge \text{studentOf}(\text{Mary}, p))$
8. $\forall x(\text{confused}(x) \rightarrow \forall y(\text{studentOf}(y, x) \rightarrow \text{confused}(y)))$
9. $\forall x \forall y((\text{studentOf}(x, y) \wedge \text{confused}(y)) \leftrightarrow (\text{confused}(x) \wedge \text{professor}(y)))$
10. $\exists p1 \exists p2(\text{professor}(p1) \wedge \text{professor}(p2) \wedge \text{confused}(p1) \wedge \text{confused}(p2) \wedge \text{studentOf}(\text{Peter}, p1) \wedge \text{studentOf}(\text{Peter}, p2) \wedge p1 \neq p2)$

2 Reasoning in first-order logic (5 Points)

Consider the following two formulas F, G in first-order logic over the signature of the first exercise:

- $F = \forall X\, \mathrm{confused}(X) \Longrightarrow \exists Y\, (\mathrm{confused}(Y) \land \mathrm{studentOf}(X, Y))$
- $G = \forall X\ \exists Y\ (\mathrm{confused}(X) \land \mathrm{confused}(Y)) \Longrightarrow \mathrm{studentOf}(X, Y)$

Show that these formulas are not equivalent by finding a model of G that is not a model of F.

Solution:

Let the domain of the model be {a, b}, and let the interpretation of the predicate symbols be:

- professor(a)
- studentOf(a, b)
- confused(a)
- confused(b)

Note that in this model, there is only one professor, which is denoted by 'a'. 'b' is not a professor, but is a confused person.

Now, let's check if this model satisfies F and G:

- F: ∀X confused(X) ==⇒ ∃Y (confused(Y) ∧ studentOf(X, Y))
    - For X = a, confused(a) holds, and there exists Y = b such that confused(b) and studentOf(a, b) hold. Therefore, F holds for X = a.
    - For X = b, confused(b) holds, but there is no Y such that confused(Y) and studentOf(b, Y) both hold. Therefore, F does not hold for X = b.
    - Since F does not hold for all X in the domain, the model does not satisfy F.
- G: ∀X ∃Y (confused(X) ∧ confused(Y)) ==⇒ studentOf(X, Y)
    - For X = a and Y = b, both confused(a) and confused(b) hold, and studentOf(a, b) also holds. Therefore, G holds for this model.

Since the model satisfies G but not F, we have shown that there exists a model of G that is not a model of F, which proves that the two formulas are not equivalent.

3 Getting familiar with Prolog (5 Points)

In this exercise, we extend the family example from the lecture slides with two new relationships:

1. Siblings have a parent in common. (That technically includes half siblings).
2. An aunt is a female sibling of a parent.

(a) Extend the lecture's rule base by rules for two new prolog predicates siblingOf/2 and auntOf/2 that capture these definitions.

The /2 after the predicate name indicates the arity of the predicate to be developed. You are allowed to use all the predicates from the lecture in your definitions, as well as the build-in predicate X \= Y stating that the instantiations of X and Y must be different.

(b) Now we extend the lecture's data base by:

- childOf(caroline, berta).
- childOf(carl, berta).
- childOf(michael, claudia).
- female(caroline).

State the result of the query ?- auntOf(X,Y).

Hint: Part (b) should actually be something that you anyways do voluntarily during part (a): When developing your rules, asking Prolog to show you all assignments that satisfy siblingOf(X,Y) and auntOf(X,Y), respectively, will help you find bugs in your definitions and improve your solution.

Solution

(a)

```
childOf(maria, claudia).
childOf(claudia, berta).
childOf(caroline, berta).
childOf(carl, berta).
childOf(michael, claudia).

female(caroline).
female(maria).
female(claudia).
female(berta).

grandchildOf(X,Z) :- childOf(X,Y), childOf(Y,Z).
mother(X) :- female(X), childOf(_,X).
male(X) :- not(female(X)).

siblingOf(X,Y):-childOf(X,Z), childOf(Y,Z), X\==Y.
auntOf(X,Y):-female(X), siblingOf(X,Z), childOf(Y,Z).
```

(b)

```
?- auntOf(X,Y)

X = caroline,
Y = maria

X = caroline,
Y = michael
```

Important hints

- Always include all names of all group members that helped solving the exercises. **on your PDF**. Only those will receive points for solving the exercises.
- By handing in this sheet, you confirm that you solved these exercises yourself. If the situation occurs that two groups have identical solutions, both groups will get zero points.
- Your SVN-Repositories can be accessed via

  https://svn.uni-koblenz.de/mhorbach/ai23/[yourGroupName]

  You can use the subfolder `workspace` to share data among your group; this folder's content will not be graded. Submit your solution in the subfolder `solutions` with speaking name such as `assignment2.pdf`. You tutor will upload their notes in the subfolder `comments`.

  Note that you do not have access to the repository's base directory.
- **Format:** All solutions must be contained in PDF documents (including source code). Additionally, source code must be provided as plain files that are readable via a standard text editor.
- Please make sure that all your programs can be run without errors. Comments on your source code will be in the annotated PDF that we create during exercise corrections.
- Do not use any mutated vowels or special characters in your source code. Also, do not use those or spaces in file names.