Artificial Intelligence

Assignment 5

Matthias Horbach
mhorbach@uni-koblenz.de

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

Submission until: 13.06.2020, 23:59
Tutorial on: 14.06.2020

1 Prolog (4 Points)

Here are four different ways to extend our family example with `childOf/2` to arbitrary descendants:

1. ```
   descendantOf(X,Y) :- childOf(X,Y).
   descendantOf(X,Y) :- childOf(X,Z), descendantOf(Z,Y).
   ```
2. ```
   descendantOf(X,Y) :- descendantOf(Z,Y), childOf(X,Z).
   descendantOf(X,Y) :- childOf(X,Y).
   ```
3. ```
   descendantOf(X,Y) :- childOf(X,Y).
   descendantOf(X,Y) :- descendantOf(Z,Y), childOf(X,Z).
   ```
4. ```
   descendantOf(X,Y) :- childOf(X,Z), descendantOf(Z,Y).
   descendantOf(X,Y) :- childOf(X,Y).
   ```

All those definitions are logically equivalent. Which ones work well in Prolog and which ones do not? Why?

The Prolog command `trace.` might help you.

2 Default rules (6 Points)

Represent the following sentences as default rules:

1. Spouses of Germans rarely hate soccer.
2. People usually have the same hometown as their spouse.
3. Most of the time, a person's hometown is where his or her employer is located.

Use predicate symbols $spouse/2, employer/2, german/1$ and $hate/2$, as well as function symbols $hometown/1, location/1$ and $soccer/0$.

3 Default theory (10 Points)

Let $T = (W, \Delta)$ be a default theory with $W = \emptyset$ and $\Delta = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5\}$ with

$$\delta_1 = \frac{\top \ : \ a}{a}, \quad \delta_2 = \frac{\top \ : \ \neg a, \ \neg c}{\neg a}, \quad \delta_3 = \frac{a \ : b}{b}, \quad \delta_4 = \frac{a \ : \ \neg b}{\neg b}, \quad \delta_5 = \frac{\neg a \ : \ c}{c}$$

Calculate all extensions of T using a process tree. Make sure that you both draw the tree and state the resulting extensions.

4 Extensions (ungraded)

Devise a default theory with exactly three extensions. Your theory should only use two predicate symbols $p/0$ and $q/0$.

Note: This exercise is an optional opportunity to train a little more. It will not be graded and does not count towards exam admission, but it will be discussed in the next tutorial.

Important hints

- Always include all names of all group members that helped solving the excercises **on your PDF**. Only those will receive points for solving the excercises.
- By handing in this sheet, you confirm that you solved these excercises 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 `assignment5.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 souce code will be in the annotated PDF that we create during excercise corrections.
- Do not use any mutated vowels or special characters in your source code. Also, do not use those or spaces in file names.