

Introduction to Logic Programming – WS 2023

Exercise Sheet 1

1 Organisational Matters

The completion of the exercises is not mandatory and you don't have to submit your solutions. We will solve and discuss each exercise sheet in the tutorial on Tuesday at 12:30 p.m. in the week after the exercise sheet has been handed out. This tutorial takes place in lecture hall 5H. If you have any questions regarding the exercises, use the forum in the ILIAS or send an email to Fabian Vu (fabian.vu@hhu.de)

In the tutorial at 2:30 p.m. you will be given further programming tasks which will be discussed within the group. This tutorial takes place in lecture hall 5C. During the semester you are able to request any topic to be repeated or intensified in this tutorial if necessary. For this, send an email to David Geleßus (dagel101@hhu.de).

Please bring your laptop for both tutorials if possible. The tutorials start on the 24th of October.

2 Preparation

Install the current version of the SWI-Prolog interpreter. If you already want to get familiar with Prolog, you might have a look at the 99 Prolog problems¹. The first programming tutorials will mostly discuss exercises from there.

We will provide unit tests for all programming exercises which are implemented using the PIUnit library of SWI-Prolog. Read the sections about PIUnit² of the SWI-Prolog handbook.

¹<https://sites.google.com/site/prologsite/prolog-problems>

²[http://www.swi-prolog.org/pldoc/doc_for?object=section\(%27packages/plunit.html%27\)](http://www.swi-prolog.org/pldoc/doc_for?object=section(%27packages/plunit.html%27))

3 Exercises

The exercises will be discussed on 24th October 2023.

Exercise 1 (Lecture – Inverted Classroom)

Watch the lecture video *LP3 - Prolog First Steps*³ in the HHU Mediathek. You can also watch *LP1 - Logic and Prolog*⁴ and *LP2 - Demo Session*⁵ to repeat the topics of the first lecture. The corresponding slides are uploaded in ILIAS: 1_Intro_2023.pdf and 2_PrologIntro.pdf

The complete playlist is available at: <https://mediathek.hhu.de/playlist/691>.

Note: you have to log in with your HHU account (Uni-Kennung) to see the lecture videos!

Exercise 2 (Interpretation of Propositional Formulae)

List all interpretations of the formula $a \vee (b \Rightarrow c)$ which are not a model.

Exercise 3 (Prolog Programming)

We start with a basic programming exercise which should emphasize the difference between facts and rules in Prolog.

The task is to create a system which is able to make specific statements on a population of sheeps. In a first step we want to model relationships between sheeps.

For this, we first store the gender of each sheep which is represented as a fact in Prolog:

```
1 female(haba).  
2 male(gunter).
```

In order to store relationships between sheeps, we further store the parent-child relationship as facts in Prolog. In particular, we use a fact of arity 2. That means the fact takes two arguments, which is usually written as parent/2. For instance, the following fact states that gunter is a parent of haba.

```
1 parent(gunter, haba).
```

In the following tasks your solutions should be independent of a possible extension of the knowledge base (e.g., your solutions should still work if a new sheep is added).

Please use the Prolog file that was handed out with this exercise sheet where we have already defined several facts and unit tests.

a) Implement a Prolog predicate `sheep/1` which is true for all sheeps in the database.

```
1 ?- sheep(haba).  
2 true  
3  
4 ?- sheep(no_sheep).  
5 false
```

³<https://mediathek.hhu.de/watch/4ebac9ac-bab1-4adf-87c9-3ce8173fc98a>

⁴<https://mediathek.hhu.de/watch/4951c6df-308d-4ea9-a682-b6ae88ed8f64>

⁵<https://mediathek.hhu.de/watch/4190ab50-ecff-4cf3-9dea-546badad4531>

- b) Implement two Prolog predicates `father/2` and `mother/2`. The request `father(F, C)` should be true iff F is a father of C. Analogously for `mother/2`.
- c) Implement a Prolog predicate `ancestor/2`. The request `ancestor(A, D)` should be true iff A is an ancestor of D.