UNIVERSITÄT BASEL

Thorsten Möller
Maarten Schenk - maarten.schenk@unibas.ch
Michael Plüss - m.pluess@unibas.ch

# Programming Paradigms – Prolog FS 2019

## Exercise 5 Due: 24.05.2018 23:59:59

**Upload answers** to the questions **and source code** before the deadline via courses.cs.unibas.ch. Since this is the last exercise, running programs **do not** have to be demonstrated during the exercise slots; only uploaded answers and source code will be graded. Also note, of all mandatory exercises given throughout the course, you must score at least 2/3 of their total sum of points to get accepted to the final exam.

**Modalities of work:** The exercise can be completed in groups of at the most 2 people. Do not forget to provide the full name of all group members together with the submitted solution.

**IMPORTANT:** Because you will not have to present your answers to this exercise during the exercise lesson, you will have no chance to explain your answers. Please make sure that all your answers contain detailed explanations and are thoroughly documented. If your answer is ambiguous, unclear or your code is insufficiently documented, you will not be awarded the full score.

Additionally, don't forget to include the names of all team members in your answer, otherwise your answer might be corrected twice and, as a result, be falsely identified as plagiarism.

## Question 1: Bite-sized Prolog Tasks (5 points)

For each of the following task descriptions write Prolog facts and/or rules that complete the task.

Also include an example usage of the predicate that completes the task in a separate text file or as a comment in the code.

a) Return the reversed list.

Example:
```
?- reverseList([45,2,37],X).
X = [37,2,45].
```

**(1 points)**

b) Return the cubed count of even numbers in a list.

Example:

```
?- numEvenCubed([1,2,3,4,5],X).
X = 8.
```

**(2 points)**

c) Return the nth Fibonacci number.

Example:

```
?- fibonacci(20, X).
X = 6765.
```

**(2 points)**

## Question 2: Knowledge Base (5 points)

The following is a list of family members with additional information about their relation and gender:

```
Karl: Max, Henry, Sandra. Jana: Max, Henry, Sandra.
Heinrich: Patrick, Emily, Fred, Leo. Sandra: Patrick, Emily, Fred.
Leo: Doris. Lea: Doris, Hanna. Sara: Susy, Peter. Max: Susy, Peter.

Male: Karl, Max, Henry, Heinrich, Peter, Patrick, Leo.
Female: Jana, Sara, Sandra, Susy, Emily, Lea, Doris, Hanna.
```

Names on the right side of the colon are children of the person on the left. At the bottom you have a list of their respective gender.

a) Write a Prolog knowledge base containing the above information.

It should be possible to check the persons gender and who their child is.

**Hint:** You have to write three simple predicates.

**(2 points)**

b) Now that you got this knowledge base we want to use it:

```
- does Hanna have a sister? If yes what is her name?
- who is Emily's grandfather?
- what is the name of Leo's mother?
- what is the name of Henry's brother?
- who are the children of Sandra and Heinrich?
- who are Jana's grandchildren?
```

Be carefull this knowledge base is not all-knowing. But still give always the full answer to the questions as Prolog code, even if it evaluates to false very early. Do not write any new predicates, work only with those from the first part. You are allowed to use write/1.

**(3 points)**

## Question 3: Modified Zebra-Puzzle (8 points)

The Zebra-Puzzle is a well-known logic puzzle, (https://en.wikipedia.org/wiki/Zebra_Puzzle). It's slightly modified such that solution is not generally known.

There are five houses in a street ordered from left (first house) to right (last house), each of the five houses is painted in a different color and their inhabitants are of different nationalities, drink different beverages, own different pets and have different favorite programming languages. The following constraints defines the puzzle:

```
1.  The Austrian lives in the black house.
2.  The Italian  uses as his favorite programming language Haskell.
3.  Tea is drunk in the yellow house.
4.  The Frenchman drinks Water.
5.  The yellow house is immediately to the right of the purple house.
6.  The person with a horse likes to use Python.
7.  The person with a bear lives inside the blue house.
8.  Milk is drunk in the middle house.
9.  The German lives in the first house.
10. The person who has a dog lives in the house next
to the person who likes to use Java.
11. The person who has a bear lives next to the person who likes to use C.
12. The person who has a fish likes to drink coffee.
13. The Swiss has a cat.
14. The German lives next to the white house.
```

Questions: Who drinks orange juice? And who uses Prolog?

To answer this two questions, you will have to write a Prolog program that can solve this riddle.

At first you should have a look at the Prolog library called "clpfd" (http://www.swi-prolog.org/pldoc/man?section=clpfd), it's about "Constraint Logic Programming over Finite Domains".

Write a predicate puzzle/0 that solves this Zebra-Puzzle and prints out all values for the houses like below.

```
1-Color 1                   2-Color 2                ...
1-Nationality 1             2-Nationality 2          ...
1-Drink 1                   2-Drink 2                ...
1-Pet                       2-Pet 2                  ...
1-Programming Language 1    2-Programming Language2  ...
```

**Hint:** The arithmetic ("#="), the membership (ins/2) and the combinatorial constraint (all_distinct/1) and the enumeration predicate (label/1) from the "clpdf" library and flatten/2, sort/2 and format/2 might be useful to solve this question.

## Question 4: Caesar's Cipher (5 points)

A Caesar cipher (https://en.wikipedia.org/wiki/Caesar_cipher) is a very simple type of encryption where every letter is replaced by the letter a fixed number of letters down the alphabet.

The following text was shifted with a rotation of 13:

```
ebznarf rhag qbzhf!
```

- Write a predicate decipher/2 that can decipher this text.
- Do not shift punctuation marks or spaces, only shift letters.
- It is enough to only work with lowercase letters.