
Programming Paradigms – Prolog

FS 2022

Exercise 4

Due: 31.05.2022 23:55:00

Upload your answers to the questions **and source code** on Adam before the deadline.

Text : For answers to questions, observations and explanations, we suggest writing them in LaTeX. Please hand-in your answers as a **single PDF** file (independent of what tools you use, LaTeX, Markdown etc.).

Source-Code : For coding exercises, the source-code must be provided and has to be **commented in detail** (e.g. how it works, how it is executed, comments on conditions to be satisfied).

Upload : Please archive multiple files into a **single compressed zip-file**. If you upload an updated version of your solutions, the file name should contain a clear and intuitive versioning number. Only the latest version will be graded.

Requisit : In order to take the final exam, you must score at least $\frac{2}{3}$ of all available points throughout the mandatory exercises.

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.

Question 1: Bite-sized Prolog Tasks

(7 points)

Write Prolog facts and/or rules that complete each of the following tasks. 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) List-Manipulation

1. Define a list with elements 3, 5, 7, 9.

1	?- lst(X).
2	X = [3, 5, 7, 9].

2. Calculate the length of a list.

```
1 ?- listLength([45, 2, 37], X).  
2 X = 3.
```

3. Reverse the order of a list.

```
1 ?- rev([1, 2, 3, 4], X).  
2 X = [4, 3, 2, 1].
```

4. Return and remove the first element of a list.

```
1 ?- pop([1, 2, 3, 4], X, Y).  
2 X = 1,  
3 Y = [2, 3, 4].
```

(4 points)

b) Check whether a number is prime by returning its largest prime factor.

```
1 ?- largestPrimeFactor(13, X).  
2 X = 13.
```

(2 points)

Question 2: Knowledge Base

(5 points)

The following listing contains all the bones of a human body, sorted by area and type¹.

Spine (vertebral column)

- Cervical vertebrae
 - atlas
 - axis / epistropheus
 - C3 ... C6
 - vertebra prominens
- Thoracic vertebrae
 - T1 ... T12
- Lumbar vertebrae
 - L1 ... L5 (L6)
- Sacrum
 - S1 ... S5
- Coccygeal vertebrae / Caudal
 - C1 ,, C4

Chest (thorax)

- Sternum
- Ribs
 - left-right
 - R1-R7
 - R8-R10 false
 - R11-R12 floating

Pelvis (fused)

- Ilium
- Ischium
- Pubis

¹List based on https://en.wikipedia.org/wiki/List_of_bones_of_the_human_skeleton completed by multiple public knowledge sources.

Skull

- Cranial bones
 - Occipital bone
 - Parietal bones left-right
 - Frontal bone
 - Temporal bones left-right
 - Sphenoid bone
 - Ethmoid bone
- Facial bones
 - Nasal bones left-right
 - Maxillae left-right
 - Lacrimal bone left-right
 - Zygomatic bone left-right
 - Palatine bone left-right
 - Inferior nasal concha left-right
 - Vomer
 - Hyoid bone
 - Mandible
- Middle ears
 - Malleus left-right
 - Incus left-right
 - Stapes left-right

Arm (left-right)

- upper arm bones
 - Humerus
 - pectoral girdle scapula
 - pectoral girdle clavicle
- lower arm bones
 - Ulna
 - Radius

Hand (left-right)

- Carpals
 - Scaphoid bone
 - Lunate bone
 - Triquetral bone
 - Pisiform bone
 - Trapezium bone
 - Trapezoid bone
 - Capitate bone
 - Hamate bone
- Metacarpals I ... V
- Phalanges
 - Proximal I ... V
 - Intermediate II ... V
 - Distal I ... V

Leg (left-right)

- Femur
- Patella
- Tibia
- Fibula

Foot (left-right)

- Tarsus/Tarsals
 - Calcaneus
 - Talus
 - Navicular bone
 - Medial cuneiform bone
 - Intermediate cuneiform bone
 - Lateral cuneiform bone
 - Cuboid bone
- Metatarsals I ... V
- Metatarsals
 - Proximal I ... V
 - Intermediate II ... V
 - Distal I ... V

- a) Write a Prolog knowledge base that contains at least 25 bones. Make sure to include different body parts and sub-structures.

It should be possible to check if a bone is part of a given body part and sub-structure. The body part should be part of the predicate. This allows the assignment of a bone to multiple body parts and/or sub-structures.

```
1 ?- partOf(humerus , upperArm , arm).  
2 true.
```

It should also be possible to query the knowledge base for the location of a given bone or list every bone that is part of a body part and/or substructure.

```
1 ?- partOf(humerus , X, Y).  
2 X = upperArm ,  
3 Y = arm .
```

(4 points)

- b) Explain the concept of backtracking in Prolog. How is it responsible for finding a complete list of bones for a given body part and/or sub-structure?

(1 points)

Question 3: Unification

(5 points)

What will Prolog return for the following terms? Briefly explain why each term can be unified or not.

1. X .
2. $3 * 6 = *(3, 6)$.
3. $1 = [X|Y]$.
4. $[1, 7, X, 9] = [X|[Y|Z]]$.
5. Only 2 out of 3 work, but why?
 - a) $\text{play_with}(\text{Bob}, X) = \text{play_with}(\text{Rita}, X)$.
 - b) $\text{play_with}(\text{Rita}, \text{play_with}(\text{Bob}, X)) = \text{play_with}(\text{Rita}, X)$.
 - c) $\text{play_with}(\text{Bob}, X) = \text{play_with}(X, \text{Rita})$.

Question 4: Operation Puzzle

(8 points)

The input for this math puzzle is a matrix of numbers. The goal is to insert mathematical operations inbetween adjacent values (horizontal and vertical). Operations are executed element-wise from left to right and top to bottom. A further rule is that each row and column must contain at least one equality. Allowed operations are: addition, subtraction, multiplication and equality. Your task is to write a program that assigns operations inbetween neighboring elements without producing a contradiction.

Example:

```
1  -? op_puzzle ([2, 3, 5, 7, 11],
2                [4, 2, 1, 4, 1],
3                [6, 6, 6, 28, 12]).
4
5      2 -   3 +   5 +   7 = 11
6      +     *     +     *     *
7      4 +   2 *   1 -   4 =   2
8      =     =     =     =     =
9      6 =   6 =   6 = 28 - 22
10
11     or
12
13      2 =( 3 *   5 +   7)/ 11
14      +     *     *     *     *
15      4 =( 2 *   1 *   4)/   2
16      =     =     =     =     =
17      6 -   6 -   6 + 28 = 22
18
19     or
20
21     ...
```

Note :

- Parenthesis were only added to improve readability. Operations are applied one by one (left-to-right, top-to-bottom).
 - To simplify the output, a simple list of equations is fine (no matrix-representation required).
- a) Write a rule that checks whether a given sequence of numbers and operators is mathematically correct.
- (1 points)
- b) Write a query that finds a solution for a given $(1 \times N)$ matrix (single row).
- (3 points)
- c) Write a query that finds **all** solutions for a given $(1 \times N)$ matrix (single row).

(3 points)

- d) Write an extended version of the query that returns all solutions for a given $(N \times M)$ matrix.

(1 points)

- e) Bonus Task :

Write an extended version of the query that finds all permutations of any 2D matrix, allowing for other operators such as power-of and divide-and-invert.

(0 points)