Programming Paradigms Exercise 3 - Prolog 2

Marco Montali Thomas Tschager

2nd Semester 2017/18

1. (a) Implement a program in Prolog to find the minimal element in a list of numbers. For example, query `?- min_elem([19,3,29],X).` should produce the output `X=3`.

   (b) List all the steps of the execution given the list [5,2,9].

   (c) Query the program as follows:
   `?- min_elem([19,3,29],X).`
   and, after getting the first answer, use `;` to get other solutions. What happens? Why? How could you use cut (`!`) so as to fix this behaviour and, more in general, to simplify the solution?

2. Write a program in Prolog that reverses a list. For example, if you input `[a,b,c,d,e]` you would get `[e,d,c,b,a]` as output.
   Hint: You can use the built-in predicate `append(X,Y,Z)` for lists. In this predicate `Z` is a concatenation of `X` and `Y`. It can be used to check if `X` concatenated `Y` is `Z` or to construct `Z` given `X` and `Y`.

3. (a) Define a binary predicate `make_flat` in Prolog that flattens a nested list. For example, query `?- make_flat([a,[b,c],[d,[e,f]]],X).` would bind `X` to list `[a,b,c,d,e,f]`.
   Hint: think about the solution declaratively, discriminating between the case where you encounter an element from the case where you encounter a (nested) list. You can again take advantage from the `append` predicate.

   (b) Query the program as follows:
   `?-make_flat(X,[a,b,c]).`
   What are the answers produced by this program, in declarative terms? What does Prolog actually return?

4. Define a predicate `dom(X,Y)` which is true iff a list `X` is not equal to `Y` and all components of `X` are less or equal than in `Y`. No components are therefore higher and at least one component is lower. For example, `dom([1,3,5],[1,4,8])` is `true`, but `dom([1,3,5],[1,3,5])` and `dom([4,3,5],[1,4,8])` are `false`.

5. Define a predicate `firstPrimeBetween(A,B,P)` which is true *iff* `P` is the smallest prime number between `A` and `B`. For example, the following query:

   ```
   ?- firstPrimeBetween(12,24,X).
   ```

   returns `X = 13` and stops.

6. There is a tariff system for public transport where prices are based on traveled kilometers (using buses, trains and cable cars). The price for each kilometer depends on how many kilometers were traveled before, as shown in the following table:

previously traveled kilometers	price
0 up to 1000	0,08 EUR/km
1001 up to 10000	0,04 EUR/km
10001 up to 20000	0,02 EUR/km
more than 20000	0,00 EUR/km

   Write a Prolog program which computes the cost to travel `X` kms.
   Hint: first think about how to encode the table, then how to encode the calculation mechanism (with a classical recursive scheme).