Exercise 1

Foundations of Logic Programming

Dr. Paola Bruscoli, Amin Timany
International Master in Computational Logic – winter term 2011/2012

Exercise 1.1
Using the Prolog program from Slides, give the answer for the following queries:

- ?-connection(frankfurt, X).
- ?-connection(X, maui).

Exercise 1.2
Define a Prolog predicate \texttt{mul} for multiplication. (You may want to use the predicate \texttt{add} defined in the slides.) Using your program give the outputs for the following queries:

- ?-mul(s(s(0), Z)).
- ?-mul(s(s(0)), s(s(s(s(s(0)))))).

Exercise 1.3
Use the definition in the previous exercise to define the factorial (\texttt{fact}) predicate.

Example ? – \texttt{fact}(s(s(s(0))), F). should produce the result: \( F = s(s(s(s(s(s(0)))))) \).

Exercise 1.4
Define a predicate \texttt{palindrome} which checks whether a a list \( L \) is a palindrome (i.e. \( L \) reverse of \( L \) are identical) or not.

Example ? – \texttt{palindrome}([a, b, c, b, a]). should produce the result: \texttt{yes}.

Exercise 1.5
Define a predicate \texttt{deletem}(K, L, Elem, RL) which deletes the \( K \)-th first elements from list \( L \) and returns the deleted elements as \( Elem \) and reduced list \( RL \).

Example ? – \texttt{deletem}(4, [3, 4, 5, 4], Elem, RL). should produce the result: \( Elem = 4 \) and \( RL = [3, 4, 5] \).

Exercise 1.6
Define a predicate \texttt{sorted}(List) which checks if a list of integers is stored in weak ascending order (i.e. each member is less than or equal to the next.)

Example ? – \texttt{sorted}([3, 4, 4, 5]). should produce the result: \texttt{yes}.