Southern Denmark University
IMADA
DM552, Part II

2017-2018

Chapter 3:
Prolog (Lists, Arithmetic, Operators)

Exercise 01: Write the following predicates in Prolog?

1- Return the maximum of 2 numbers.
2- Return the maximum of 3 numbers.
3- Determine the sum of 2 numbers.
4- Determine the sum of N numbers.
5- Check whether a given number is pair or not.
6- Determine the **n!** of the given number n.

Exercise 02:
Consider a given list L, write the following prolog predicates:

1- Find the first element of L.
2- Find the last element of L.
3- Find an element x in the list L.
4- Compute the number of elements in L.
5- Return the sum of elements of L
6- Find the number of occurrences of an element X in L.
7- Check if the list L1 is the sub-List of L2
8- Return the inverse list of L.
9- Compare the elements of two lists

Exercice03:

The aim of this exercise is to compute the basic operations of Boolean algebra in Prolog. Assume that : $a \Rightarrow b \equiv \neg a$ **or** b ; $a \Leftrightarrow b \equiv a \Rightarrow b$ **and** $b \Rightarrow a$ et $a \oplus b \equiv \neg a$ **and** b **or** a **and** $\neg b$

1- Write the following facts in Prolog :

or (x, y, z) : z is the disjunction of x and y.

and (x, y, z) : z is the conjunction of x and y.

negation(x, y) : y is the negation of x.

2- Write in Prolog the following rules:

Implication (a, b, c): c is the result of the implication between a and b.

Equivalence (a, b, c): c is the result of the equivalence between a and b.

Xor(a, b, c) : c is the results of Xor between a and b.

3- Interrogate the program prolog by responding to these two following request:
$X \cdot Y \equiv Y \cdot X$
$X + Y \equiv Y + X$

Exercice04:
A palindrome is a list that can be read forward or backward, for instance, the list [x, a, m, a, x]. Write Prolog program that find out whether a list is a palindrome?

Exercice05:
The aim of this exercise is to compress the given list. If a list contains repeated elements they should be replaced with a single copy of the element. The order of the elements should not be changed.

Example:
?- compress([a,a,a,a,b,c,c,a,a,d,e,e,e,e],X).
X = [a,b,c,a,d,e]

Write Prolog program that compress a list L1 into another list L2: compress(L1, L2)?

Exercice06:

Use the result of the previous exercise to implement an improved compressing function taking into account the number of elements. ?

Example:
?- CompressImprovedVersion([a,a,a,a,b,c,c,a,a,d,e,e,e,e],X).
X = [[4,a],[1,b],[2,c],[2,a],[1,d],[4,e]]