

Functional and logic programming

- written exam -

Important:

1. Subjects are graded as follows: of - 1p; A – 1.5p; B - 2.5p; C - 2.5p; D - 2.5p.
2. Prolog problems will be resolved using SWI Prolog. The following are required: (1) explanation of the code and of the reasoning behind it; (2) recursive model that solves the problem, for all the predicates used; (3) specification of every predicate (parameters and their meaning, flow model, type of the predicate - deterministic/non-deterministic).
3. Lisp problems will be resolved using Common Lisp. The following are required: (1) explanation of the code and of the reasoning behind it; (2) recursive model that solves the problem, for each function used; (3) specification of every function (parameters and their meaning).

A. Let L be a list of numbers and given the following PROLOG predicate definition with flow model (i, o):

$f([], 0).$

$f([H|T], S) :- \underline{f(T, S1)}, S1 \geq 2, !, S \text{ is } S1 + H.$

$f([_|T], S) :- \underline{f(T, S1)}, S \text{ is } S1 + 1.$

Rewrite the definition in order to avoid the recursive call $\underline{f(T, S)}$ in both clauses. Do NOT redefine the predicate. Justify your answer.

B. Given a nonlinear list that contains numerical and non-numerical atoms, write a Lisp program that verifies if the following three lists are equal: the list of all atoms on levels multiple of 3 (3, 6, etc.), the list of all atoms on levels of the form $3k+1$ (1, 4, 7, etc.) and the list of all atoms on levels of the form $3k+2$ (2, 5, 8, etc.). For example, for the list (A 1 (A 1(A 1(B 777 (B (B 777 C) 777 C) C) D) D) D) the result will be true.

C. Write a PROLOG program that generates the list of all combinations of k elements with numbers from 1 to N , with the property that difference between two consecutive numbers from a combination has an even value. Write the mathematical models and flow models for the predicates used. For example, for the $\mathbf{N}=4$, $\mathbf{k}=2 \Rightarrow [[1,3],[2,4]]$ (not necessarily in this order).

D. An n-ary tree is represented in Lisp as (node subtree1 subtree2 ...). Write a Lisp function to verify whether a node **x** occurs on an even level of the tree. The root level is assumed zero. **A MAP function shall be used.**

Example for the tree (a (b (g)) (c (d (e)) (f)))

a) $x=g \Rightarrow T$ **b)** $x=h \Rightarrow \text{NIL}$