

Examen PP – Seria CC

16.06.2017

Timp de lucru 2 ore . 100p necesare pentru nota maximă

1. Determinați forma normală pentru următoarea expresie, ilustrând pașii de reducere:
 $((\lambda z.\lambda y.\lambda x.(y\ z)\ y)\ \lambda y.y)$ 15p
2. Este vreo diferență (ca efect, la execuție) între cele două linii de cod Racket? Dacă da, care este diferența?; dacă nu, de ce nu diferă?
`(let ((a 1) (b 2)) (+ a b))`
`((lambda (a b) (+ a b)) 1 2)` 15p
3. Implementați în Racket funcția `f` care primește o listă și determină elementul mai mare decât modulul oricărui alt element. Folosiți, în mod obligatoriu, cel puțin o funcțională. 15p
4. Sintetizați tipul funcției `f` (în Haskell): `f g h l1 l2 = filter (g . h) (l1 ++ l2)` 15p
5. Scrieți definiția în Haskell a clasei `Ended` care, pentru un tip colecție `t` construit peste un alt tip `v`, definește o funcție `frontEnd` care extrage primul element din colecție și o funcție `backEnd` care extrage ultimul element din colecție.
Instanțiați această clasă pentru tipul `data Triple a = T a a a` 15p
6. Știind că *Ucenicul vrea sa învețe pe dascăl*, și că *ucenic(Luke) și dascal(Yoda)*, demonstrați folosind rezoluția că *vrea-să-învețe(Luke, Yoda)* este adevărat . 15p
7. Implementați în Prolog predicatul `x(L, M)` care detemină, pentru o listă `L`, `M`, minimumul listei. Nu folosiți recursivitate explicită. 15p
8. Implementați un algorim Markov care primește în șirul de intrare un număr binar și scade 2 din acest număr. Exemple: `10 - 10 = 0`; `100 - 10 = 10`; `111 - 10 = 101`; `10001 - 10 = 111` Este ok dacă numărul rezultat începe cu 0. 15p