

UNIVERSITÀ DEGLI STUDI DELLA CALABRIA
Corso di Laurea Magistrale in Ingegneria Informatica
(TRACCIA A)
Prova d'esame di *Linguaggi per L'Intelligenza Artificiale*
(durata della prova: 120 minuti)

Haskell

Esercizio 1

Si scriva la definizione di una funzione che presi in input un intero x e una lista di interi xs restituisce la lista degli interi in xs diversi da x mantenendo però la prima occorrenza di x (elimina quindi da xs le ripetizioni di x). L'ordine di apparizione degli interi deve essere mantenuto.

`delete :: Int -> [Int] -> [Int]`

esempio: `delete 1 [2,1,3,4,1,6]` restituisce `[2,1,3,4,6]`

Esercizio 2

Definire la funzione

`estraisottoliste :: Int -> [Int] -> [[Int]]`

che ricevendo in input un intero k e una lista di interi xs restituisce una lista contenente tutte le sottoliste contigue di xs di lunghezza k

esempio: `estraisottoliste 2 [1, 2, 3]` è uguale a `[[1, 2], [2, 3]]`

Esercizio 3

Qual è il tipo dell'espressione definita nel seguente modo

`map (+1)`

Prolog

Esercizio 4

Definire il predicato `posizionen(N, ListaIn, ListaOut)` che, presa una lista di elementi ground, `ListaIn`, restituisce la lista composta dagli elementi di `ListaIn` che si trovano in posizione multipla di N (assumiamo che il primo elemento di una lista occupi la posizione 1).

Esempio: `posizionen(3, [a, b(c), 7, [1,2,3], pippo, pluto, paperino], ListaOut)` deve unificare `ListaOut` con `[7, pluto]`.

Esercizio 5

Si considerino i seguenti predicati

`rally(nome)`

`vincitore(nomepilota, nomerally, data)`

`pilota(nome)`

si definisca il seguente predicato prolog:

`rallyvinti(nomepilota, valore)` # che restituisce vero se `nomepilota` ha vinto `valore` rally

`piubravo(nomepilota)` # che restituisce vero se `nomepilota` ha vinto il maggior numero di rally

`piùcontinuo(nomepilota)` # che restituisce vero se `nomepilota` ha vinto il maggior numero di rally consecutivi

Esercizio 6

Quale tra i seguenti predicati implementa il not (inteso non come negazione logica ma come incapacità di provare un goal a partire dalle informazioni contenute nella base di conoscenza di Prolog)? Motivare la risposta.

`not1(X) :- X, !, fail.`

`not1(X).`

`not2(X) :- !, X, fail.`

`not2(X).`

`not3(X) :- X, fail, !.`

not3(X).