Laurea Magistrale in INFORMATICA
Principi di Linguaggi di Programmazione
Paradigmi

prof. M. Bellia
Appello IV - june 25th, 2013

(Timing: 2 hours – Grading: (pts n-m) is the score range to be obtained in each exercise)

Exercise 1. (pts. 5 - 9) Let H be the set of finite strings on {a,b,c,d}. Use Prolog for defining:
(a) (pts 1) A concrete representation of H values;
(b) (pts 4) A binary predicate anag/2(x,y) which holds only if x and y are in H and x is an anagram of v;
(c) (pts 4) A predicate split/4(u,w1,w2,w3) which holds only if u, w1, w2, w3 are in H and, u=w1.w.w3, i.e. u is the juxtapposition of w1, w, w3, and w2 is an anagram of w.
(Define all the auxiliary predicates that are used in Your solution)

Exercise 2. (pts. 5 - 10) Let iTree be an ADT for immutable trees with nodes of a generic type and of arbitrary outdegree. The iTree's have the following public operations:

- *newEmpty()*: returns the empty iTree;
- *newVertex*(r): returns the iTree with the only root r, i.e. without sons;
- *addEdge*(t,r,s): returns the iTree t' which differs from t, at most for the edge, (r,s), namely with root r and a new son s, provided it can be added. Otherwise an exception is raised.
- *sons*(t,r): returns the list of the nodes of t that are sons of r.

Use Caml for defining:

(a) (pts 2) The API for iTree;
(b) (pts 8) An ADT for iTree such that:
it includes a private operation *inList*(l,r) that:
  i. returns true iff the list l contains the value v of a suitable type for l;
  ii. it is defined by using the iterative programming methodology

Exercise 3. (pts. 6 - 11) Let
Sia mTree una classe Java per un ADT di alberi, come quelli in esercizio 2, ma ora, modificabili.

(a) (pts 4) Si definisca una classe Java, mTreeS, che estenda mTree aggiungendo una nuova operazione pubblica *fanOut()* che calcola il massimo outdegree dei nodi dell'albero.
(b) (pts 7) Si definisca una classe Java, mTreeR, che estenda MTreeS aggiungendo una nuova operazione pubblica *remove(r,s)* che modifica l'albero rimuovendo l'arco (r,s), se possibile. In caso contario solleva eccezione.