

Linguaggi di Programmazione con Laboratorio
Corso di Laurea in Matematica
Appello del 18 Luglio 2018

Nota. Per superare la prova occorre svolgere tutti gli esercizi, riportando su ciascuno il punteggio minimo di 15/30

Tempo a disposizione 2h.15'

Esercizio1 (Tempo Atteso per lo Svolgimento = 30')

Si considerino le Api, NODE, EDGE, GRAPH, nell'allegato, che introducono i tipi 'a node, ('a,'b) edge, ('a,'b) graph, per nodi etichettati su un generico 'a, archi orientati etichettati su un generico 'b, e grafi con nodi e archi di tali tipi. Si assumano dati i relativi moduli Adt, Node, Edge che forniscono un codice OCaml per l'uso dei valori ['a node] e [('a,'b) edge].

Si fornisca in OCaml, un modulo Adt, Graph, che implementi il modulo GRAPH, limitandosi a mostrare:

- a. **Intestazione, struttura** del modulo, **Stato Concreto, AF ed I.**
- b. Definizione delle operazioni: mk, addE, isEin.
- c. Definizione degli additional: copy, eq.

e delle eventuali, operazioni ausiliarie relative alla definizione. delle operazioni in (b) e (c).

Esercizio2 (TAS = 45')

Una **foresta** è un'unione di alberi tra loro disgiunti e può essere ottenuta da un grafo diretto quando l'indegree¹ di ogni suo nodo è al massimo 1: Le radici (degli alberi) della foresta sono i nodi del grafo aventi indigree 0. Gli archi uscenti da ogni nodo della foresta sono ordinati in accordo all'ordine in cui sono stati inseriti nel grafo.

Si assumano definite in Java, le classi Node<A> ed Edge<A,B> per Adt degli stessi valori astratti **IMMUTABLE**, visti in esercizio1, e Graph<A,B> per Adt di grafi diretti dei corrispondenti valori **MUTABLE**.

Si definisca in Java una nuova classe **Forest<A,B>** che estenda **Graph<A,B>** e la specializzi per soli valori foresta **MUTABLE**. In particolare, si fornisca:

- a. **Intestazione, Struttura** di classe, **Stato Concreto, AF ed I** di **Forest<A,B>**;
- b. **Overriding** di tutte le operazioni di **Graph<A,B>** che lo richiedano
- c. Definizione dei metodi:

roots che restituisce un **Vector<Node<A>>** con tutte e sole le radici della foresta;
getSons che, dato un nodo, restituisce **Vector<Node<A>>** con le radici di tutti i sottoalberi del nodo, nell'ordine atteso.

e degli eventuali, metodi ausiliari relativi alla loro definizione.

¹ indegree è il numero di archi entranti in un nodo.