Principi di Linguaggi di Programmazione
Programming Paradigms

prof. M. Bellia
Exam III - June18th, 2014
(Time to do it: 2 hours – Mandatory: Get one half at least, of the total points assigned to each exercise)

Exercise 1 (pts 10)

a) Provide a definition, in a language of Your choice, of the function, *fib*, computing the n-th of the Fibonacci series and using
   a) tail-recursive technique;
   b) memoization technique;
   Comments the codes.

b) How many invocations of fib are needed in order to compute the expressions fib(2) and fib(4) in this order, in the case (a) and in the case (b), respectively?

Exercise 2 (pts 10)

Let *graph* be an abstract data type for immutable graphs in Ocaml. Nodes are labelled by distinct values of a generic type. Edges are directed and labelled by values of a generic type. The type *graph* has the following public operations

empty() returns an empty graph;
add(g,x,y,u) returns a graph that differs from g for at most one edge, if not already in g, from x to y and labelled by u.
edges(g) returns a list of distinct pairs which contains the pair (x,y) if and only if g has at least one edge from x to y.
outgoing(g,x) returns a list of distinct pairs which contains the pair (y,u) if and only if g contains an edge from x to y and labelled by u.

Noting that multi-digraphs are allowed, i.e. graphs may contains more that one edge from x to node y, provided they are labelled by different values.

(a) Show the Ocaml API of the type graph;
(b) Show one Ocaml ADT for the API;
(c) Let *equals: graph * graph -> boolean* be a predicate that returns true if and only the two arguments are the same graph. Provide a definition of equals.

Exercise 3 (pts 10)

(a) Show a Java abstract class *Graph* for the API of Exercise 2.a, above, except that now the graphs are mutable values.
(b) Let *GraphADT* be an extension of Graph providing an implementation for the ADT. Show a class *GraphADTE* that extends *GraphADT* with an additional public operation *equals* that computes analogously to the one in 2.c above.
(c) Show a class *GraphADTA* that extends *GraphADT* on acyclic graphs and has an additional public operation:
reached returns a LinkedList containing all the nodes that are reached from the argument.