A.A. 2003-04

Linguaggi di Programmazione

Progetto (III "esonero")

data di assegnazione: 13 maggio 2004
data di consegna: 1 giugno 2004

Inferenza dei tipi in SML

1. Definire in SML un datatype `term` che implementi la sintassi del linguaggio *Fun* con annotazioni esplicite dei tipi. Scrivere un programma SML che esegua il type checking dei termini di `term`.
2. Scrivere un programma SML che implementi l'inferenza dei tipi sui termini di *Fun* (*senza* annotazioni esplicite dei tipi).

Hint

```
(* Il linguaggio Fun
   ----------------- *)

datatype term = BOOL of bool |
                INT of int |
                VAR of string |
                FN of string * term |
                APPL of term * term |
                LET of string * term * term;


(* Esempio: il termine
fn x => (let val y = (fn z => x y) in z y end)
si scrive come segue: *)

val t = FN("x", LET("y",
                    FN("z", APPL(VAR("x"), VAR("y"))),
                    APPL(VAR("z"), VAR("y"))));
```

```
(* la funzione isfree ha tipo string * term -> bool.
   isfree(x,M) controlla se x occorre libera in M. *)

fun isfree (_, BOOL(_)) = false
  | isfree (_, INT(_)) = false
  | isfree (x, VAR(y)) = (x = y)
  | isfree (x, FN(y,M)) = (x <> y) andalso (isfree(x,M))
  | isfree (x, APPL(M,N)) = isfree(x,M) orelse isfree(x,N)
  | isfree (x, LET(y,M,N)) = isfree(x,M) orelse
                             ((x <> y) andalso isfree(x,N));
```