

## Problem 4 (Exam 049161 E98)

```
infix Add datatype expr = Const of int
                          | Var of string
                          | Add of expr * expr;

infix PAdd datatype pat = PConst of int
                          | PVar of string
                          | PAdd of pat * pat;

(* A *)
fun vars(PVar s)          = [s]
  | vars(p1 PAdd p2)     = vars p1 @ vars p2
  | vars _                = [];

(* B *)
infix member
fun y member xs = List.exists (fn x => x=y) xs;

fun duplicate []          = false
  | duplicate(x::xs)     = x member xs orelse duplicate xs;

fun legal p = not (duplicate(vars p));

(* C *)
type binding             = string * expr
type bindingList        = binding list;

local
  exception Pat;
  fun m(PConst x, Const x') = if x=x' then [] else raise Pat
    | m(PVar s, e)           = [(s,e)]
    | m(p1 PAdd p2, e1 Add e2) = m(p1,e1) @ m(p2,e2)
    | m _                    = raise Pat
in
  fun match p e = if legal p then SOME(m(p,e)) else NONE
    handle Pat => NONE
end;
```