(\* type variables \*)

type 'a t = 'a list

let f (a : 'a list) : 'a = List.hd a

(\* polymorphic variants \*)

type t = [ `A | `B ]

(\* variants \*)

type result = Sat | Unsat | Unknown

(\* module and module types \*)

module type S = sig

val compute : unit -> unit

end

module Impl : S = struct

let compute () = ()

end