1 Foundations of Computer Science (LCP)

This question has been translated from Standard ML to OCaml

(a) Write brief notes on polymorphism in OCaml, using lists and standard list functions such as \texttt{append} and \texttt{List.map}. [4 marks]

(b) Explain the meaning of the following declaration and describe the corresponding data structure, including the role of polymorphism.

\texttt{type 'a se = Void | Unit of 'a | Join of 'a se * 'a se}

[4 marks]

(c) Show that OCaml lists can be represented using this variant type by writing the functions \texttt{encode\_list} of type \texttt{'a list -> 'a se} and \texttt{decode\_list} of type \texttt{'a se -> 'a list}, such that \texttt{decode\_list (encode\_list xs) = xs} for every list \texttt{x}.

[3 marks]

(d) Consider the following function declaration:

\begin{verbatim}
let rec cute p = function
  | Void -> false
  | Unit x -> p x
  | Join (u, v) ->
      cute p u || cute p v
\end{verbatim}

What does this function do, and what is its type? [4 marks]

(e) Consider the following expression:

\begin{verbatim}
fun p -> cute (cute p)
\end{verbatim}

What does it mean, and what is its type? Justify your answer carefully. [5 marks]