2 Foundations of Computer Science (LCP)

This question has been translated from Standard ML to OCaml

(a) Write brief notes on programming with lazy lists in OCaml. Your answer should include the definition of a polymorphic type of infinite lazy lists, a function to return the tail of a lazy list, a function to create the infinite list of all positive integers, and an apply-to-all functional analogous to the list functional \texttt{map}.

[6 marks]

(b) Write a function \texttt{diag} that takes a lazy list of lazy lists, 

\[
\left[\begin{array}{c}
  z_{11}; z_{12}; z_{13}; \\
  z_{21}; z_{22}; z_{23}; \\
  z_{31}; z_{32}; z_{33};
\end{array}\right]; \\
\left[\begin{array}{c}
  \vdots
\end{array}\right]
\]

(∗) 

and returns the diagonal, namely the lazy list \(z_{11}; z_{22}; z_{33}; \ldots\). [3 marks]

(c) Write a function that takes two lazy lists \([x_1; x_2; x_3; \ldots]\) and \([y_1; y_2; y_3; \ldots]\) and a function \(f\) of two arguments; it should return a lazy list of lazy lists like (∗) above, with \(z_{ij} = f(x_i, y_j)\). [3 marks]

(d) Write a function that converts a lazy list of lazy lists like (∗) above to a lazy list whose elements are all of the \(z_{ij}\), enumerated in some order. [8 marks]