

COMPUTER SCIENCE TRIPOS Part IA – 2017 – Paper 1

2 Foundations of Computer Science (LCP)

This question has been translated from Standard ML to OCaml

(a) Define an OCaml variant type for infinite lists, without the possibility of finite lists. Briefly illustrate programming techniques for your variant type by declaring

(i) a recursive functional (analogous to `map` for ordinary lists) that applies a given function to every element of an infinite list.

(ii) a function for generating infinite lists of the form $x, f(x), f(f(x)), \dots$ for any given f and x .

[6 marks]

(b) Briefly explain why the function analogous to `append (@)` for ordinary lists is of no value for your infinite list data type. Code a function to combine two arbitrary infinite lists, yielding a result that includes the elements of both lists.

[3 marks]

(c) Use the functions declared in your previous solutions to express an infinite list consisting of all numbers of the form $5^i \times 7^j \times 9^k$ for integers $i, j, k \geq 0$.

[3 marks]

(d) The list `[1; 5; 7; 25; 9; 35; 35; ...]` is a legitimate solution to part (c) above, but note that the integers are out of order. Code a function to merge two ordered infinite lists, and use that to modify your previous solution to yield the same set of numbers but in strictly increasing order. Briefly comment, with justification, on whether merge sort for ordinary lists can be generalised to infinite lists.

[8 marks]

For full credit, code should be concise and clear.