COMPUTER SCIENCE TRIPOS Part II – 2026 – Paper 8

2 Algebraic Techniques for Programming (nk480)

Suppose we have a string API with the following constant-time operations:

```
length : string -> int

head : string -> char   (* the first char of a nonempty string *)
tail : string -> string (* a substring omitting the first char *)

last : string -> char   (* the last char of a nonempty string *)
init : string -> string (* the substring omitting the last char *)
```

We can use them to define the function `pal` for computing the length of the longest palindrome which is a subsequence of the input `s`. (Recall that s_1 is a subsequence of s_2 if s_1 can be found by deleting elements of s_2.)

```
let rec pal s =
  if len s <= 1 then
    len s
  else if head s = last s then
    2 + pal (init(tail s))
  else
    max (pal (tail s)) (pal (init s))
```

(*a*) Give an ML datatype and map function implementing the functor corresponding to the recursion scheme of `pal`. [4 marks]

(*b*) Give a coalgebra and algebra implementing the divide and conquer phases of this algorithm. [5 marks]

(*c*) Prove that the coalgebra is well-founded. [5 marks]

(*d*) Give a type signature and non-memoizing implementation for the `hylo` recursion combinator, and then use it to implement `pal`. [2 marks]

(*e*) Estimate the worst-case complexity of the naive algorithm, and then give a bound for the version with a memoizing fixed point. Carefully explain your logic. [4 marks]