COMPUTER SCIENCE TRIPOS Part II – 2026 – Paper 9

13 Types (nk480)

(*a*) Give the type of the Church numerals in System F. [2 marks]

(*b*) Give an existential type that models the following OCaml module interface:

```
module type NatList = sig
  type t

  val nil : t
  val cons : nat -> t -> t
  val fold : t -> 'a -> (nat -> 'a -> 'a) -> 'a
end
```

[3 marks]

(*c*) Give a term with the existential type you defined in Part (*b*), as a term in System F extended with existential types and products. [5 marks]

(*d*) In the typed lambda calculus augmented with state but no monadic control, write a function $f : ((\mathbb{N} \to 1) \to 1) \to (\mathbb{N} \to 1) \to \mathbb{N}$. The function call $f\ k\ g$ should return n, where n is the number of times that g is invoked by k in the function call $k\ g$. [3 marks]

(*e*) A dependent type theory is both a system of formal logic and a programming language. Consider the following piece of Agda code, where `List A` is the type of lists with element type `A`.

```
X : (P : List A → Set) →
    (P []) →
    ((x : A) → (xs : List A) → P xs → P (x :: xs)) →
    (xs : List A) → P xs
X P base step [] = base
X P base step (x :: xs) = step x xs (X P base step xs)
```

(*i*) Explain what this function is, understood as a theorem of logic. [3 marks]

(*ii*) Explain what this function is, understood as a functional program. [4 marks]