14 Types (NK)

(*a*) Give typing rules for the introduction form $\mathsf{pack}(\tau, M)$ and elimination form $\mathsf{unpack}\ M\ \mathsf{as}\ (x, \alpha)\ \mathsf{in}\ N$ of the existential type $\exists\alpha(\tau)$.

[4 marks]

(*b*) An infinite stream of booleans can be represented in the polymorphic lambda calculus using the existential type

$$\mathsf{stream} \triangleq \exists\alpha(\alpha \times (\alpha \to (\mathsf{bool} \times \alpha)))$$

(*i*) Using the encoding above, define a function $\mathsf{head} : \mathsf{stream} \to \mathsf{bool}$. [3 marks]

(*ii*) Using the encoding above, define a function $\mathsf{tail} : \mathsf{stream} \to \mathsf{stream}$. [3 marks]

(*iii*) Using the encoding above, define a function

$$\mathsf{unfold} : \forall\alpha(\alpha \to (\alpha \to (\mathsf{bool} \times \alpha)) \to \mathsf{stream})$$

[4 marks]

(*iv*) Using unfold and the other functions above, define a function $\mathsf{notstream} : \mathsf{stream} \to \mathsf{stream}$, which returns a stream containing the boolean negation of the elements of the input stream. This answer should not use explicit pack or unpack expressions. [6 marks]

[*Note:* You may use extensions to the pure polymorphic lambda calculus such as let-bindings, natural numbers, products, and sum types, but carefully note their use and their typing in your answers.]