15   Types (nk480)

(*a*) Consider System F extended with existential types, products, and a natural number type.

   (*i*) Give an existential type corresponding to an abstract type of booleans with constructors for true and false, as well as a conditional test (if-then-else) operation. [3 marks]

   (*ii*) Give an implementation of this type, using the natural numbers as the representation of booleans. [4 marks]

(*b*) Suppose we extend the simply-typed lambda calculus with the ability to raise exceptions with the fail construct, and the ability to catch exceptions with the try e_0 except e_1 construct. Suppose also that we track exceptions monadically, with the type Exn A representing possibly-failing computations of A.

   (*i*) Give a typing rule for signalling an error with fail. [2 marks]

   (*ii*) Give a typing rule for trapping an error with try e_0 except e_1. Does your type for this term have an effect? Justify your design. [5 marks]

(*c*) Consider the simply-typed lambda calculus extended with natural numbers and reference types, but without monadic effect tracking.

   (*i*) Give an expression of type $1 \rightarrow \mathbb{N}$, which evaluates to a function which counts the number of times it has been called. [2 marks]

   (*ii*) Without using explicit recursion, give an expression and its type in the simply-typed lambda calculus with references whose execution never halts. [4 marks]