13 Types (nk480)

(*a*) Derive the following entailments with the natural deduction system for classical logic.

(*i*) Show that $A \vee B; A \vdash B\,\mathsf{true}$. [5 marks]

(*ii*) Show that $A \vee B, \neg A; \cdot \vdash B\,\mathsf{true}$. [7 marks]

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

(*i*) Give a Church encoding for an optional natural number type (corresponding to `nat option` in OCaml). [2 marks]

(*ii*) Give an existential type corresponding to an abstract type of optional naturals, with constructors for `Some` and `None`, as well as a case analysis operation. It should correspond to the following OCaml module signature:

```
module type ONAT = sig
  type t
  val none : t
  val some : nat -> t
  val case : t -> 'a -> (nat -> 'a) -> 'a
end
```

[3 marks]

(*iii*) Give an implementation of this existential type. [3 marks]