EB5 - Universal and Existential Types

Exercise 1

Considering the Church encoding of booleans seen in class (*i.e.CBool*), complete the following exercises.

I. Given the operation *not* defined in class, evaluate the terms:

a) $not\ True$

b) $not\ (not\ True)$

II. Encode (and exhibit their behavior through examples) the following operations on *CBool*:

a) Conjunction

b) Implication

c) If-then-else

Exercise 2

Encode the function $isZero : CNat \to CBool$ and exhibit its behavior by applying it to a sample Church numeral.

Exercise 3

Provide terms with the following types:

I. $\forall X.\forall Y.\forall Z.(X \to Y \to Z) \to (Y \to X \to Z)$ (reverse order of arguments)

II. $\forall X.\forall Y.\forall Z.(X \to Y) \to (Y \to Z) \to (X \to Z)$ (function composition)

Exercise 4

Suppose we have type constructors for product $\sigma \times \tau$ and sum $\sigma + \tau$ with the usual syntax. Moreover, suppose the terms for building expressions of these types are encoded externally (functional approach) in System F.

$$\begin{array}{rcl} M & ::= & \ldots\ldots \mid \langle M, M\rangle \mid fst(M) \mid snd(M) \\ & \mid & inl(M) \mid inr(M) \mid case(M)\ of\ \{x \mapsto M; y \mapsto M\} \end{array}$$

provide terms with the following types:

I. $\forall X.\forall Y.\forall Z.(X \to Z) \to (Y \to Z) \to (X + Y \to Z)$

II. $\forall X.\forall Y.\forall Z.(X + Y \to Z) \to (X \to Z)$

III. $\forall X.\forall Y.\forall Z.(X \times Y \to Z) \to (X \to Y \to Z)$ (currying)

IV. $\forall X.\forall Y.\forall Z.(X \to Y \to Z) \to (X \times Y \to Z)$ (uncurrying)

Exercise 5

Assume the following internal encoding of the sum type in System F:

$$CSum\ X\ Y \stackrel{\text{def}}{=} \forall R.(X \to R) \to (Y \to R) \to R$$

Provide terms in System F that encode the following operations:

I. *inl* that injects a value into the left of a sum.

II. *inr* that injects a value into the left of a sum.

III. *case* that performs case analysis on values of sum types as follows:

$$\frac{M \to M'}{case\ M\ of\ x \mapsto P; y \mapsto Q \to case\ M'\ of\ x \mapsto P; y \mapsto Q}$$

$$\frac{}{case\ (inl\ V)\ of\ x \mapsto P; y \mapsto Q \to P\{x \leftarrow V\}}$$

$$\frac{}{case\ (inr\ V)\ of\ x \mapsto P; y \mapsto Q \to Q\{y \leftarrow V\}}$$

Exercise 6

Type and evaluate: $cons\ [\mathbb{N}]\ 1\ (cons\ [\mathbb{N}]\ 2\ nil)$

Exercise 7

Evaluate the following expressions:

I. $isNil\ nil$

II. $isNil\ (cons\ [\mathbb{N}]\ 1\ (cons\ [\mathbb{N}]\ 2\ nil))$

III. $head\ [\mathbb{N}]\ (cons\ [\mathbb{N}]\ 1\ nil)$

Exercise 8

Given the encoding of pairs seen in class:

$$CPair\ X\ Y\ = \forall R.(X \to Y \to R) \to R$$

define the projection operations fst and snd.

Exercise 9

Consider the following notion of binary trees:

```
type BinTreeNat = Leaf | Node of Nat*BinTreeNat*BinTreeNat
```

These trees can be encoded as the following type:

$$CBinTreeNat \stackrel{\text{def}}{=} \forall R.(\mathbb{N} \to R \to R) \to R \to R$$

Encode the following operations:

I. $leaf : CBinTreeNat$

II. $node : \mathbb{N} \to CBinTreeNat \to CBinTreeNat \to CBinTreeNat$

III. $flip : CBinTreeNat \to CBinTreeNat$ that returns the mirror image

Exercise 10

Provide type derivations for the expressions:

I. $m1 = \langle \mathbb{N}, \{a = 0, f = lx : \mathbb{N}.succ(x)\}\rangle$

II. $m2 = \langle \mathbb{B}, \{a = True, f = lx : \mathbb{B}{,}0\}\rangle$

III. $let\ \{X, x\} = m1\ in\ (x.f\ x.a)$

IV. $let\ \{X, x\} = m2\ in\ (x.f\ x.a)$

Exercise 11

Define the ADT of stacks of natural numbers as a module. The supported operations should include: new, push, pop, top, isEmpty. Use lists to implement these stacks.