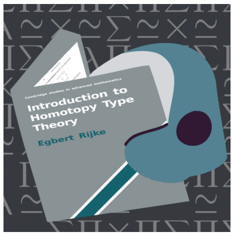

Worksheet 4

HoTTEST Summer School 2022

The HoTTEST TAs, and

11 July 2022

1 $(\star)$

We define the standard finite types $\mathsf{Fin} : \mathbb{N} \to \mathsf{Type}$ inductively with constructors

$$\mathsf{pt} : \Pi_{n:\mathbb{N}} \mathsf{Fin}(\mathsf{suc}(n))$$
$$\mathsf{i} : \Pi_{n:\mathbb{N}} \mathsf{Fin}(n) \to \mathsf{Fin}(\mathsf{suc}(n)).$$

Spell out all elements of $\mathsf{Fin}(3)$.

It is common practice to leave the argument n of the constructors implicit. Then the induction principle states that a dependent function

$$f : \Pi_{n:\mathbb{N}} \Pi_{x:\mathsf{Fin}(n)} P_n(x)$$

is determined by

$$g_n : \Pi_{x:\mathsf{Fin}(n)} P_n(x) \to P_{\mathsf{suc}(n)}(\mathsf{i}(x))$$

and

$$p_n : P_{\mathsf{suc}(n)}(\mathsf{pt}).$$

The function f satisfies the judgemental equalities

$$f_{\mathsf{suc}(n)}(\mathsf{i}(x)) \doteq g_n(x, f_n(x))$$
$$f_{\mathsf{suc}(n)}(\mathsf{pt}) \doteq p_n.$$

2 $(\star\star\star)$

It is also possible to define the standard finite types $\mathsf{Fin'} : \mathbb{N} \to \mathsf{Type}$ recursively as a type family over $\mathbb{N}$,

$$\begin{aligned}\mathsf{Fin'}(0) &\doteq \emptyset \\ \mathsf{Fin'}(\mathsf{suc}(n)) &\doteq \mathsf{Fin'}(n) + \mathbb{1}.\end{aligned}$$

We suggestively use the notation $\mathsf{i'} : \mathsf{Fin'}_n \to \mathsf{Fin'}_{\mathsf{suc}(n)}$ and $\mathsf{pt'} : \mathsf{Fin'}_{\mathsf{suc}(n)}$ for the inclusions inl and inr into the coproduct $\mathsf{Fin'}(n) + \mathbb{1}$. Formulate the induction principle of $\mathsf{Fin'}$.

3 $(\star\star)$

Choose your favourite version of the finite types. Use pattern matching to define two different inclusions $\iota, \hat{\iota} : \Pi_{n:\mathbb{N}}\mathsf{Fin}(n) \to \mathbb{N}$, such that the images of $\iota_{\mathsf{suc}(n)}$ and $\hat{\iota}_{\mathsf{suc}(n)}$ are the first $n+1$ natural numbers.

4 $(\star)$

Give a recursive definition of the ordering relation $\leq : \mathbb{N} \to \mathbb{N} \to \mathsf{Type}$.

5 $(\star\star)$

Define $\mathsf{is\text{-}prime} : \mathbb{N} \to \mathsf{Type}$.

6 (★★)

State the twin prime conjecture and Goldbach's conjecture in HoTT.

7 (★★)

Suppose we had constructed a proof

$$\mathsf{infinitude\text{-}of\text{-}primes} : \Pi_{n:\mathbb{N}}\Sigma_{p:\mathbb{N}}(\mathsf{is\text{-}prime}(p) \times (n \leq p)).$$

Further assume that the prime p returned by this program is the least prime above n. A definition of such a term can be found in the Agda UniMath library[1]. Construct a function $\mathsf{prime} : \mathbb{N} \to \mathbb{N}$ which computes the n-th prime.

8 (★★)

We define the predicate

$$\mathsf{is\text{-}decidable}(A) \doteq A + \neg A$$

for an arbitrary type A. Do we expect

$$\Pi_{n:\mathbb{N}}\mathsf{is\text{-}decidable}(\mathsf{is\text{-}prime}(n))$$

to be true (inhabited)? Why or why not?

[1] https://unimath.github.io/agda-unimath/elementary-number-theory.infinitude-of-primes.html

9 $(\star\star\star)$

Suppose we had a proof

$$\mathsf{is\text{-}decidable\text{-}is\text{-}prime} : \Pi_{n:\mathbb{N}}\mathsf{is\text{-}decidable}(\mathsf{is\text{-}prime}(n)).$$

Construct a function

$$\mathsf{prime\text{-}counting} : \mathbb{N} \to \mathbb{N}$$

which computes the number of primes less than or equal to its input.

10 $(\star\star\star)$

Show that adding k is an injective function which respects equality, i.e. that

$$(m = n) \leftrightarrow (m +_{\mathbb{N}} k = n +_{\mathbb{N}} k)$$

for all $m, n, k : \mathbb{N}$.