



Worksheet 4

HoTTEST Summer School 2022

The HoTTEST TAs, and
11 July 2022

1 (★)

We define the standard finite types $\mathbf{Fin} : \mathbb{N} \rightarrow \mathcal{U}_0$ inductively with constructors

$$\begin{aligned} \mathbf{pt} &: \prod_{n:\mathbb{N}} \mathbf{Fin}(\mathbf{succ}(n)) \\ \mathbf{i} &: \prod_{n:\mathbb{N}} \mathbf{Fin}(n) \rightarrow \mathbf{Fin}(\mathbf{succ}(n)). \end{aligned}$$

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

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

$$f : \prod_{n:\mathbb{N}} \prod_{x:\mathbf{Fin}(n)} P_n(x)$$

is determined by

$$g_n : \prod_{x:\mathbf{Fin}(n)} P_n(x) \rightarrow P_{\mathbf{succ}(n)}(\mathbf{i}(x))$$

and

$$p_n : P_{\mathbf{succ}(n)}(\mathbf{pt}).$$

The function f satisfies the judgemental equalities

$$\begin{aligned} f_{\mathbf{succ}(n)}(\mathbf{i}(x)) &\doteq g_n(x, f_n(x)) \\ f_{\mathbf{succ}(n)}(\mathbf{pt}) &\doteq p_n. \end{aligned}$$

2 (★ ★ ★)

It is also possible to define the standard finite types $\mathbf{Fin}' : \mathbb{N} \rightarrow \mathcal{U}_0$ recursively as a type family over \mathbb{N} ,

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

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

3 (★★)

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

4 (★)

Give a recursive definition of the ordering relation $\leq : \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathcal{U}_0$.

5 (★★)

Define $\mathbf{is\text{-}prime} : \mathbb{N} \rightarrow \mathbf{Type}$.

6 (★★)

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

7 (★★)

Suppose we had constructed a proof

$$\text{infinitude-of-primes} : \prod_{n:\mathbb{N}} \Sigma_{p:\mathbb{N}} (\text{is-prime}(p) \times (\text{suc}(n) \leq_{\mathbb{N}} p)).$$

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 library¹. Construct a function $\text{prime} : \mathbb{N} \rightarrow \mathbb{N}$ which computes the n -th prime.

8 (★★)

We define the predicate

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

for an arbitrary type A . Do we expect

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

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

9 (★ ★ ★)

Suppose we had a proof

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

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

Construct a function

`prime-counting` : $\mathbb{N} \rightarrow \mathbb{N}$

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