

# Chapter 1

## Type theory

### Exercises

*Exercise 1.1.* Given functions  $f : A \rightarrow B$  and  $g : B \rightarrow C$ , define their **composite**  $g \circ f : A \rightarrow C$ . Show that we have  $h \circ (g \circ f) \equiv (h \circ g) \circ f$ .

**Solution** (Alan).

**Solution** (Alex).

*Exercise 1.2.* Derive the recursion principle for products  $\text{rec}_{A \times B}$  using only the projections, and verify that the definitional equalities are valid. Do the same for  $\Sigma$ -types.

**Solution** (Daniel).

**Solution** (Jake).

**Solution** (James).

*Exercise 1.3.* Derive the induction principle for products  $\text{ind}_{A \times B}$ , using only the projections and the propositional uniqueness principle  $\text{uniq}_{A \times B}$ . Verify that the definitional equalities are valid. Generalize  $\text{uniq}_{A \times B}$  to  $\Sigma$ -types, and do the same for  $\Sigma$ -types. (*This requires concepts from ??.*)

**Solution** (Steven). I actually have no idea (yet), this is just a test.  $\text{foo} = b^{a^r}$

**Solution** (Zack).

*Exercise 1.4.* Assuming as given only the *iterator* for natural numbers

$$\text{iter} : \prod_{C:\mathcal{U}} C \rightarrow (C \rightarrow C) \rightarrow \mathbb{N} \rightarrow C$$

with the defining equations

$$\begin{aligned} \text{iter}(C, c_0, c_s, 0) &:\equiv c_0, \\ \text{iter}(C, c_0, c_s, \text{succ}(n)) &:\equiv c_s(\text{iter}(C, c_0, c_s, n)), \end{aligned}$$

derive a function having the type of the recursor  $\text{rec}_{\mathbb{N}}$ . Show that the defining equations of the recursor hold propositionally for this function, using the induction principle for  $\mathbb{N}$ .

**Solution** (Alan).

**Solution** (Daniel).

*Exercise 1.5.* Show that if we define  $A + B \equiv \sum_{(x:\mathbf{2})} \text{rec}_2(\mathcal{U}, A, B, x)$ , then we can give a definition of  $\text{ind}_{A+B}$  for which the definitional equalities stated in ?? hold.

**Solution** (Steven).

**Solution** (Alex).

**Solution** (James).

*Exercise 1.6.* Show that if we define  $A \times B \equiv \prod_{(x:\mathbf{2})} \text{rec}_2(\mathcal{U}, A, B, x)$ , then we can give a definition of  $\text{ind}_{A \times B}$  for which the definitional equalities stated in ?? hold propositionally (i.e. using equality types). (This requires the function extensionality axiom, which is introduced in ??.)

**Solution** (Jake).

**Solution** (Zack).

*Exercise 1.7.* Give an alternative derivation of  $\text{ind}'_{=A}$  from  $\text{ind}_{=A}$  which avoids the use of universes. (This is easiest using concepts from later chapters.)

**Solution** (Alan).

**Solution** (Jake).

*Exercise 1.8.* Define multiplication and exponentiation using  $\text{rec}_{\mathbb{N}}$ . Verify that  $(\mathbb{N}, +, 0, \times, 1)$  is a semiring using only  $\text{ind}_{\mathbb{N}}$ . You will probably also need to use symmetry and transitivity of equality, ????

**Solution** (Steven).

**Solution** (Alex).

*Exercise 1.9.* Define the type family  $\text{Fin} : \mathbb{N} \rightarrow \mathcal{U}$  mentioned at the end of ??, and the dependent function  $\text{fmax} : \prod_{(n:\mathbb{N})} \text{Fin}(n+1)$  mentioned in ??.

**Solution** (Daniel).

**Solution** (Zack).

**Solution** (James).

*Exercise 1.10.* Show that the Ackermann function  $\text{ack} : \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N}$  is definable using only  $\text{rec}_{\mathbb{N}}$  satisfying the following equations:

$$\begin{aligned} \text{ack}(0, n) &\equiv \text{succ}(n), \\ \text{ack}(\text{succ}(m), 0) &\equiv \text{ack}(m, 1), \\ \text{ack}(\text{succ}(m), \text{succ}(n)) &\equiv \text{ack}(m, \text{ack}(\text{succ}(m), n)). \end{aligned}$$

**Solution** (Alan).

**Solution** (Steven).

*Exercise 1.11.* Show that for any type  $A$ , we have  $\neg\neg\neg A \rightarrow \neg A$ .

**Solution** (Jake).

**Solution** (Daniel).

**Solution** (James).

*Exercise 1.12.* Using the propositions as types interpretation, derive the following tautologies.

- (i) If  $A$ , then (if  $B$  then  $A$ ).
- (ii) If  $A$ , then not (not  $A$ ).
- (iii) If (not  $A$  or not  $B$ ), then not ( $A$  and  $B$ ).

**Solution** (Alex).

**Solution** (Zack).

*Exercise 1.13.* Using propositions-as-types, derive the double negation of the principle of excluded middle, i.e. prove *not (not ( $P$  or not  $P$ ))*.

**Solution** (Alan).

**Solution** (Zack).

*Exercise 1.14.* Why do the induction principles for identity types not allow us to construct a function  $f : \prod_{(x:A)} \prod_{(p:x=x)} (p = \text{refl}_x)$  with the defining equation

$$f(x, \text{refl}_x) :\equiv \text{refl}_{\text{refl}_x} \quad ?$$

**Solution** (Daniel).

**Solution** (Alex).

**Solution** (James).

*Exercise 1.15.* Show that indiscernability of identicals follows from path induction.

**Solution** (Steven).

**Solution** (Jake).

*Exercise 1.16.* Show that addition of natural numbers is commutative:  $\prod_{(i,j:\mathbb{N})} (i + j = j + i)$ .