Chapter 1

Type theory

Exercises

Exercise 1.1. Given functions $f : A \to B$ and $g : B \to C$, define their **composite** $g \circ f : A \to C$. 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 $\mathsf{rec}_{A\times B}$ using only the projections, and verify that the definitional equalities are valid. Do the same for Σ-types.

Solution (Daniel)**.** The *recursion principle* is the rule that states that $\mathsf{rec}_{A\times B}(f) : A \times B \to C$ is well defined, and it is taken as a primitive notion. We are asked to instead take the well-definedness of pr_1 and pr_2 as primitive instead and to derive the recursion principle.

Definitions of $\mathsf{rec}_{A\times B}$, pr_1, and pr_2 are given below for convenience.

$$\begin{aligned}
&\mathsf{rec}_{A\times B} : (A \to B \to C) \to A \times B \to C\\
&\mathsf{rec}_{A\times B}(f)((a, b)) := f(a)(b)
\end{aligned}$$

$$\begin{aligned}
&\mathsf{pr}_1 : A \times B \to A\\
&\mathsf{pr}_1 := \mathsf{rec}_{A\times B}(\lambda\, a\, b \mapsto a)
\end{aligned}$$

$$\begin{aligned}
&\mathsf{pr}_2 : A \times B \to B\\
&\mathsf{pr}_2 := \mathsf{rec}_{A\times B}(\lambda\, a\, b \mapsto b)
\end{aligned}$$

(In agreement with the notation used in the text, $\lambda\, a\, b \mapsto \Phi$ is merely shorthand for $\lambda\, a \mapsto (\lambda\, b \mapsto \Phi)$. We can think of such an expression as a "two-variable" curried function [mmmm, curry].)

We need primitive definitions of pr_1 and pr_2.

$$\mathsf{pr}_1 : A \times B \to A$$
$$\mathsf{pr}_1(a, b) := a$$

$$\mathsf{pr}_2 : A \times B \to B$$
$$\mathsf{pr}_2(a, b) := b$$

We define

$$\mathsf{rec}'_{A\times B} : (A \to B \to C) \to A \times B \to C$$
$$\mathsf{rec}'_{A\times B}(f)((a, b)) := f(\mathsf{pr}_1(a, b))(\mathsf{pr}_2(a, b))$$

We have

$$\begin{aligned}\mathsf{rec}_{A\times B}(f)(a, b) &= f(a)(b)\\ &= f(\mathsf{pr}_1(a, b))(\mathsf{pr}_2(a, b))\\ &= \mathsf{rec}'_{A\times B}(f)(a, b)\end{aligned}$$

So $\mathsf{rec}_{A\times B} = \mathsf{rec}'_{A\times B}$.

Solution (Jake)**.**

Solution (James)**.**

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

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

Solution (Zack)**.**

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

$$\mathsf{iter} : \prod_{C:\mathcal{U}} C \to (C \to C) \to \mathbb{N} \to C$$

with the defining equations

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

derive a function having the type of the recursor $\mathsf{rec}_{\mathbb{N}}$. 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 $A+B :\equiv \sum_{(x:\mathbf{2})} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)$, then we can give a definition of 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 $A \times B :\equiv \prod_{(x:\mathbf{2})} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)$, then we can give a definition of $\mathsf{ind}_{A\times B}$ 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 $\mathsf{ind}'_{=_A}$ from $\mathsf{ind}_{=_A}$ 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 $\mathsf{rec}_{\mathbb{N}}$. Verify that $(\mathbb{N}, +, 0, \times, 1)$ is a semiring using only $\mathsf{ind}_{\mathbb{N}}$. You will probably also need to use symmetry and transitivity of equality, **????**.

Solution (Steven)**.**

Solution (Alex)**.**

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

Solution (Daniel)**.**

Solution (Zack)**.**

Solution (James)**.**

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

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

Solution (Alan)**.**

Solution (Steven)**.**

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

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 $f : \prod_{(x:A)} \prod_{(p:x=x)} (p = \mathsf{refl}_x)$ with the defining equation

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

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: $\prod_{(i,j:\mathbb{N})} (i + j = j + i)$.