Homotopy Type Theory

Induction

5.1 Introduction to inductive types

T 5.1.1. *Let $f, g : \prod_{(x:\mathbb{N})} E(x)$ be two functions which satisfy the recurrences*

$$e_z : E(0) \qquad \text{and} \qquad e_s : \prod_{n:\mathbb{N}} E(n) \to E(\mathsf{succ}(n))$$

up to propositional equality, i.e., such that

$$f(0) = e_z \qquad \text{and} \qquad g(0) = e_z$$

as well as

$$\prod_{n:\mathbb{N}} f(\mathsf{succ}(n)) = e_s(n, f(n)),$$

$$\prod_{n:\mathbb{N}} g(\mathsf{succ}(n)) = e_s(n, g(n)).$$

Then f and g are equal.

5.2 Uniqueness of inductive types

5.3 W-types

T 5.3.1. *Let $g, h : \prod_{(w:\mathsf{W}_{(x:A)} B(x))} E(w)$ be two functions which satisfy the recurrence*

$$e : \prod_{a,f} \Big(\prod_{b:B(a)} E(f(b)) \Big) \to E(\mathsf{sup}(a, f)),$$

propositionally, i.e., such that

$$\prod_{a,f} g(\mathsf{sup}(a, f)) = e(a, f, \lambda b.\, g(f(b))),$$

$$\prod_{a,f} h(\mathsf{sup}(a, f)) = e(a, f, \lambda b.\, h(f(b))).$$

Then g and h are equal.

5.4 Inductive types are initial algebras

D 5.4.1. A $\mathbb{N}$**-algebra** is a type C with two elements $c_0 : C$, $c_s : C \to C$. The type of such algebras is

$$\mathbb{N}\mathsf{Alg} :\equiv \sum_{C:\mathcal{U}} C \times (C \to C).$$

D 5.4.2. A $\mathbb{N}$**-homomorphism** between $\mathbb{N}$-algebras (C, c_0, c_s) and (D, d_0, d_s) is a function $h : C \to D$ such that $h(c_0) = d_0$ and $h(c_s(c)) = d_s(h(c))$ for all $c : C$. The type of such homomorphisms is

$$\mathbb{N}\mathsf{Hom}((C, c_0, c_s), (D, d_0, d_s)) :\equiv \sum_{(h:C \to D)} (h(c_0) = d_0) \times \textstyle\prod_{(c:C)} (h(c_s(c)) = d_s(h(c))).$$

D 5.4.3. A $\mathbb{N}$-algebra I is called **homotopy-initial**, or **h-initial** for short, if for any other $\mathbb{N}$-algebra C, the type of $\mathbb{N}$-homomorphisms from I to C is contractible. Thus,

$$\mathsf{isHinit}_{\mathbb{N}}(I) :\equiv \prod_{C:\mathbb{N}\mathsf{Alg}} \mathsf{isContr}(\mathbb{N}\mathsf{Hom}(I, C)).$$

T 5.4.4. *Any two h-initial $\mathbb{N}$-algebras are equal. Thus, the type of h-initial $\mathbb{N}$-algebras is a mere proposition.*

T 5.4.5. *The $\mathbb{N}$-algebra $(\mathbb{N}, \mathbf{0}, \mathsf{succ})$ is homotopy initial.*

T 5.4.6. *For any type $A : \mathcal{U}$ and type family $B : A \to \mathcal{U}$, the W-algebra*

Notes

Exercises

ex Derive the induction principle for the type $\mathsf{List}(A)$ of lists from its
definition as an inductive type in **??**.

Exer. 5.1. Construct two functions on natural numbers which satisfy the same recurrence (e_z, e_s) judgmentally, but are not judgmentally equal.

Exer. 5.2. Construct two different recurrences (e_z, e_s) on the same type E which are both satisfied judgmentally by the same function $f : \mathbb{N} \to E$.

Exer. 5.3. Show that for any type family $E : \mathbf{2} \to \mathcal{U}$, the induction operator

$$\mathsf{ind}_{\mathbf{2}}(E) : (E(0_{\mathbf{2}}) \times E(1_{\mathbf{2}})) \to \prod_{b:\mathbf{2}} E(b)$$

is an equivalence.

Exer. 5.4. Show that the analogous statement to **??** for $\mathbb{N}$ fails.

Exer. 5.5. Show that if we assume simple instead of dependent elimination for W-types, the uniqueness property (analogue of **??**) fails to hold. That is, exhibit a type satisfying the recursion principle of a W-type, but for which functions are not determined uniquely by their recurrence.

Exer. 5.6. Suppose that in the "inductive definition" of the type C at the beginning of **??**, we replace the type $\mathbb{N}$ by $\mathbf{0}$. Analogously to (5), we might consider a recursion principle for this type with hypothesis

$$h : (C \to \mathbf{0}) \to (P \to \mathbf{0}) \to P.$$

Show that even without a computation rule, this recursion principle is inconsistent, i.e. it allows us to construct an element of $\mathbf{0}$.

Exer. 5.7. Consider now an "inductive type" D with one constructor $\mathsf{scott} : (D \to D) \to D$. The second recursor for C suggested in **??** leads to the following recursor for D:

$$\mathsf{rec}_D : \prod_{P:\mathcal{U}} ((D \to D) \to (D \to P) \to P) \to D \to P$$

with computation rule $\mathsf{rec}_D(P, h, \mathsf{scott}(\alpha)) \equiv h(\alpha, (\lambda d.\, \mathsf{rec}_D(P, h, \alpha(d))))$. Show that this also leads to a contradiction.

Exer. 5.8. Let A be an arbitrary type and consider generally an "inductive definition" of a type L_A with constructor $\mathsf{lawvere} : (L_A \to A) \to L_A$. The second recursor for C suggested in **??** leads to the following recursor for L_A:

$$\mathsf{rec}_{L_A} : \prod_{P:\mathcal{U}} ((L_A \to A) \to P) \to L_A \to P$$

with computation rule $\mathsf{rec}_{L_A}(P, h, \mathsf{lawvere}(\alpha)) \equiv h(\alpha)$. Using this, show that A has the **fixed-point property**, i.e. for every function $f : A \to A$ there exists an $a : A$ such that $f(a) = a$. In particular, L_A is inconsistent if A is a type without the fixed-point property, such as $\mathbf{0}$, $\mathbf{2}$, or $\mathbb{N}$.

Exer. 5.9. Continuing from **??**, consider $L_{\mathbf{1}}$, which is not obviously inconsistent since $\mathbf{1}$ does have the fixed-point property. Formulate an induction principle for $L_{\mathbf{1}}$ and its computation rule, analogously to its recursor, and using this, prove that it is contractible.

Exer. 5.10. In **??** we defined the type $\mathsf{List}(A)$ of finite lists of elements of some type A. Consider a similar inductive definition of a type $\mathsf{Lost}(A)$ whose only constructor is

$$\mathsf{cons} : A \to \mathsf{Lost}(A) \to \mathsf{Lost}(A).$$

Show that $\mathsf{Lost}(A)$ is equivalent to $\mathbf{0}$.

Exer. 5.11. Suppose A is a mere proposition, and $B : A \to \mathcal{U}$.

(i). Show that $\mathsf{W}_{(a:A)} B(a)$ is a mere proposition.

(ii). Show that $\mathsf{W}_{(a:A)} B(a)$ is equivalent to $\sum_{(a:A)} \neg B(a)$.

(iii). Without using $\mathsf{W}_{(a:A)} B(a)$, show that $\sum_{(a:A)} \neg B(a)$ is a homotopy W-type $\mathsf{W}^h_{(a:A)} B(a)$ in the sense of **??**.

Exer. 5.12. Let $A : \mathcal{U}$ and $B : A \to \mathcal{U}$.

(i). Show that $\left(\sum_{(a:A)} \neg B(a)\right) \to \left(\mathsf{W}_{(a:A)} B(a)\right)$.

(ii). Show that $\left(\mathsf{W}_{(a:A)} B(a)\right) \to \left(\neg \prod_{(a:A)} B(a)\right)$.

Exer. 5.13. Let $A : \mathcal{U}$ and suppose that $B : A \to \mathcal{U}$ is decidable, i.e. $\prod_{(a:A)} (B(a) + \neg B(a))$ (see **??**). Show that $\left(\mathsf{W}_{(a:A)} B(a)\right) \to \left(\sum_{(a:A)} \neg B(a)\right)$.

Exer. 5.14. Show that the following are logically equivalent.

(i). $\left(\mathsf{W}_{(a:A)} B(a)\right) \to \left\|\sum_{(a:A)} \neg B(a)\right\|$ for any $A : \mathsf{Set}$ and $B : A \to \mathsf{Prop}$.