Homotopy Type Theory

Induction

Introduction to inductive types

T 5.0.1. *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.

Uniqueness of inductive types

W-types

T 5.0.2. *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.

Inductive types are initial algebras

D 5.0.3. 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 5.0.4. 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 5.0.5. 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 5.0.6. *Any two h-initial $\mathbb{N}$-algebras are equal. Thus, the type of h-initial $\mathbb{N}$-algebras is a mere proposition.*

T 5.0.7. *The $\mathbb{N}$-algebra $(\mathbb{N}, \mathbf{0}, \mathsf{succ})$ is homotopy initial.*

T 5.0.8. *For any type $A : \mathcal{U}$ and type family $B : A \to \mathcal{U}$, the W-algebra $(\mathsf{W}_{(x:A)}B(x), \mathsf{sup})$ is h-initial.*

Homotopy-inductive types

T 5.0.9. *For any $A : \mathcal{U}$ and $B : A \to \mathcal{U}$, the type $\mathsf{W}_d(A, B)$ is a mere proposition.*

T 5.0.10. *For any $A : \mathcal{U}$ and $B : A \to \mathcal{U}$, the type $\mathsf{W}_s(A, B)$ is a mere proposition.*

T 5.0.11. *For any $A : \mathcal{U}$ and $B : A \to \mathcal{U}$, the type $\mathsf{W}_h(A, B)$ is a mere proposition.*

L 5.0.12. *For any $A : \mathcal{U}$ and $B : A \to \mathcal{U}$, we have*

$$\mathsf{W}_d(A, B) \simeq \mathsf{W}_s(A, B) \simeq \mathsf{W}_h(A, B)$$

T 5.0.13. *The types satisfying the formation, introduction, elimination, and propositional computation rules for W-types are precisely the homotopy-initial W-algebras.*

T 5.0.14. *The rules for natural numbers with propositional computation rules can be derived from the rules for W-types with propositional computation rules.*

The general syntax of inductive definitions

NB 5.0.15. There is a question of universe size to be addressed. In general, an inductive type must live in a universe that already contains all the types going into its definition. Thus if in the definition of D, the ambiguous notation Prop means $\mathsf{Prop}_{\mathcal{U}}$, then we do not have $D : \mathcal{U}$ but only $D : \mathcal{U}'$ for some larger universe $\mathcal{U}'$ with $\mathcal{U} : \mathcal{U}'$. In a predicative theory, therefore, the right-hand side of (5) lives in $\mathsf{Prop}_{\mathcal{U}'}$, not $\mathsf{Prop}_{\mathcal{U}}$. So this contradiction does require the propositional resizing axiom mentioned in **??**.

Generalizations of inductive types

Identity types and identity systems

D 5.0.16. Let A be a type and $a_0 : A$ an element.

- A **pointed predicate** over (A, a_0) is a family $R : A \to \mathcal{U}$ equipped with an element $r_0 : R(a_0)$.
- For pointed predicates (R, r_0) and (S, s_0), a family of maps $g : \prod_{(b:A)} R(b) \to S(b)$ is **pointed** if $g(a_0, r_0) = s_0$. We have

$$\mathsf{ppmap}(R, S) :\equiv \sum_{g:\prod_{(b:A)} R(b)\to S(b)} (g(a_0, r_0) = s_0).$$

- An **identity system at** a_0 is a pointed predicate (R, r_0) such that for any type family $D : \prod_{(b:A)} R(b) \to \mathcal{U}$ and $d : D(a_0, r_0)$, there exists a function $f : \prod_{(b:A)} \prod_{(r:R(b))} D(b, r)$ such that $f(a_0, r_0) = d$.

T 5.0.17. *For a pointed predicate (R, r_0) over (A, a_0), the following are logically equivalent.*

(i). (R, r_0) *is an identity system at* a_0.
(ii). *For any pointed predicate* (S, s_0), *the type* $\mathsf{ppmap}(R, S)$ *is contractible.*
(iii). *For any* $b : A$, *the function* $\mathsf{transport}^R(-, r_0) : (a_0 =_A b) \to R(b)$ *is an equivalence.*
(iv). *The type* $\sum_{(b:A)} R(b)$ *is contractible.*

D 5.0.18. An **identity system** over a type A is a family $R : A \to A \to \mathcal{U}$ equipped with a function $r_0 : \prod_{(a:A)} R(a, a)$ such that for any type family $D : \prod_{(a,b:A)} R(a, b) \to \mathcal{U}$ and $d : \prod_{(a:A)} D(a, a, r_0(a))$, there exists a function $f : \prod_{(a,b:A)} \prod_{(r:R(a,b))} D(a, b, r)$ such that $f(a, a, r_0(a)) = d(a)$ for all $a : A$.

T 5.0.19. *For $R : A \to A \to \mathcal{U}$ equipped with $r_0 : \prod_{(a:A)} R(a, a)$, the following are logically equivalent.*

(i). (R, r_0) *is an identity system over* A.
(ii). *For all* $a_0 : A$, *the pointed predicate* $(R(a_0), r_0(a_0))$ *is an identity system at* a_0.
(iii). *For any* $S : A \to A \to \mathcal{U}$ *and* $s_0 : \prod_{(a:A)} S(a, a)$, *the type*

$$\sum_{(g:\prod_{(a,b:A)} R(a,b)\to S(a,b))} \prod_{(a:A)} g(a, a, r_0(a)) = s_0(a)$$

is contractible.
(iv). *For any* $a, b : A$, *the map* $\mathsf{transport}^{R(a)}(-, r_0(a)) : (a =_A b) \to R(a, b)$ *is an equivalence.*
(v). *For any* $a : A$, *the type* $\sum_{(b:A)} R(a, b)$ *is contractible.*

C 5.0.20 (Equivalence induction). *Given any type family*

$$D : \prod_{A,B:\mathcal{U}} (A \simeq B) \to \mathcal{U}$$

and function $d : \prod_{(A:\mathcal{U})} D(A, A, \mathsf{id}_A)$, there exists

$$f : \prod_{(A,B:\mathcal{U})} \prod_{(e:A\simeq B)} D(A, B, e)$$

such that $f(A, A, \mathsf{id}_A) = d(A)$ for all $A : \mathcal{U}$.