15-819 Homotopy Type Theory
Lecture Notes

Kristina Sojakova and Joseph Lee

November 25, 2013

1   Higher Inductive Types

Recall last week we discussed (lower) inductive types and their definitions. This week, we move on to discussing higher inductive types. Intuitively, a higher inductive type (i.e. HIT) can be seen as a type with inductive definitions along with equational laws.

A higher order type then is a generalization of a free algebraic structure, which would have generators and equational laws that should be fulfilled. For example, a group given by a set of generators and relations (e.g. commutativity).

This gives a full higher-dimensional structure which allows us to impose relations in multiple dimensions. In a proof-relevant environment, this means we get generators and more generators. Or in other words, because of proof-relevance, equational laws can be seen as generators at a higher dimension.

For example, at the 0-type level, the generators are elements or points. These are called the 0-cells. Identities between 0-cells would be 1-cells, identities between 1-cells would be 2-cells, and so on. This allows for a type to be defined over multiple dimensions by the use of n-cells.

2   The Interval Type I

Recall that the interval type I is defined by two points, 0_I and 1_I, along with an identity seg. That is, I is defined by the 0-cells

$$0_I : I$$
$$1_I : I$$

and the 1-cell

$$\mathsf{seg} : 0_I =_I 1_I$$

The recursor is defined as

$$\frac{\Gamma \vdash M : I \quad \Gamma \vdash a : A \quad \Gamma \vdash b : A \quad \Gamma \vdash \beta : a =_A b}{\Gamma \vdash \mathsf{rec}[A](a; b; \beta)(M) : A} \; I\mathsf{rec}$$

where the β_0-rules that should hold (i.e. β-rules for 0-cells) are

$$\mathsf{rec}[A](a; b; \beta)(0_I) \equiv a$$
$$\mathsf{rec}[A](a; b; \beta)(1_I) \equiv b$$

and the β_1-rule is

$$(\mathsf{ap}_{\mathsf{rec}[A](a;b;\beta)}(\mathsf{seg}) =_{a=_A b} \beta) \; \mathsf{true}$$

That is to say, that the propositional equality above is inhabited.

From here, we can define the induction principle, which is similar to the recursor:

$$\frac{\Gamma, z : I \vdash A(z) : \mathcal{U} \quad \Gamma \vdash M : I \quad \Gamma \vdash a_0 : A(0_I) \quad \Gamma \vdash a_1 : A(1_I) \quad \Gamma \vdash p : a_0 =_{\mathsf{seg}}^{z.A} a_1}{\Gamma \vdash \mathsf{ind}[z.A](a_0; a_1; p)(M) : A(M)} \; I\mathsf{ind}$$

Recall that $a_0 =_{\mathsf{seg}}^{z.A} a_1$ is defined as $\mathsf{tr}[z.A](\mathsf{seg})(a_0) =_{A(1_I)} a_1$. This is because a_0 and a_1 no longer necessarily have the same type, so we need to use "path over" to express the desired equational law.

The β_0-rules for induction are

$$\mathsf{ind}[z.A](a_0; a_1; p)(0_I) \equiv a_0$$
$$\mathsf{ind}[z.A](a_0; a_1; p)(1_I) \equiv a_1$$

and the β_1-rule is

$$(\mathsf{dap}_{\mathsf{ind}[z.A](a_0;a_1;p)}(\mathsf{seg}) =_{a_0=_{\mathsf{seg}}^{z.A} a_1} p) \; \mathsf{true}$$

There is also a unicity rule, or η-rule, that states "if a function behaves like the recursor of I, then it must be the recursor." There is a similar rule for the induction principle of I.

Exercise. *Define the η-rules for $I\mathsf{rec}$ and $I\mathsf{ind}$.*

3   The Circle Type $\mathbb{S}^1$

Another example of a higher inductive type is the circle type $\mathbb{S}^1$. The $\mathbb{S}^1$ type is defined by the 0-cell (point)

$$\mathsf{base} : \mathbb{S}^1$$

and the 1-cell (path)

$$\mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}$$

The recursor is defined as

$$\frac{\Gamma \vdash M : \mathbb{S}^1 \quad \Gamma \vdash a_0 : A \quad \Gamma \vdash l : a_0 =_A a_0}{\Gamma \vdash \mathsf{rec}[A](a_0; l)(M) : A} \; \mathbb{S}^1\mathsf{rec}$$

with the β_0-rule is

$$\mathsf{rec}[A](a_0; l)(\mathsf{base}) \equiv a_0$$

and the β_1-rule is

$$(\mathsf{ap}_{\mathsf{rec}[A](a_0;l)}(\mathsf{loop}) =_{a_0 =_A a_0} l) \; \mathsf{true}$$

The induction principle is defined as

$$\frac{\Gamma, z : \mathbb{S}^1 \vdash P(z) : \mathcal{U} \quad \Gamma \vdash M : \mathbb{S}^1 \quad \Gamma \vdash b : P(\mathsf{base}) \quad \Gamma \vdash l : b =_{\mathsf{loop}}^{z.P} b}{\Gamma \vdash \mathsf{ind}[z.P](b; l)(M) : P(M)} \; \mathbb{S}^1\mathsf{ind}$$

with β_0-rule

$$\mathsf{ind}[z.P](b; l)(\mathsf{base}) \equiv b$$

and β_1-rule

$$(\mathsf{dap}_{\mathsf{ind}[z.P](b;l)}(\mathsf{loop}) =_{b =_{\mathsf{loop}}^{z.P} b} l) \; \mathsf{true}$$

Be careful with the type of l, because it is easy to write a type that "typechecks" but is not correct. l should express that taking b around the loop path over P returns to b.

Exercise. *Define the η-rules for $\mathbb{S}^1\mathsf{rec}$ and $\mathbb{S}^1\mathsf{ind}$.*

4   Total Space of Loops as a Function from $\mathbb{S}^1$

Recall that we previously characterized the total space of paths,

$$\int \mathsf{Id}_A :\equiv \sum_{x,y:A} x =_A y$$

as being equivalent to the function type from I to A, i.e.

$$(I \to A) \simeq \int \mathsf{Id}_A$$

We can similarly characterize the total space of loops

$$\int \Omega_A :\equiv \sum_{x:A} x =_A x$$

as being equivalent to the function type from $\mathbb{S}^1$ to A. i.e.

$$(\mathbb{S}^1 \to A) \simeq \int \Omega_A$$

Proof. Define $f : (\mathbb{S}^1 \to A) \to \int \Omega_A$ as

$$f = \lambda g.\langle g(\mathsf{base}), \mathsf{ap}_g(\mathsf{loop})\rangle$$

Exercise. *Show that f has a quasi-inverse.*

$\square$

5 Suspensions

Another example of a higher inductive type is the *suspension type*, which subsumes the interval and circle types (up to homotopy). For a type $A : \mathcal{U}$, the suspension of A, denoted by $\mathsf{susp}(A) : \mathcal{U}$, is the higher inductive type given by two 0-cell constructors

$$N : \mathsf{susp}(A)$$
$$S : \mathsf{susp}(A)$$

which will be referred to as the *north* and *south* poles, respectively, and a family of 1-cell constructors

$$\mathsf{mer} : A \to (N =_{\mathsf{susp}(A)} S)$$

which can be understood as a collection of *meridians*, i.e., paths from the north to the south pole.

Based on the above data, we can deduce the appropriate recursion schema: given any other type $B : \mathcal{U}$ which "looks like the suspension of A", there should be a function, called the recursor, from $\mathsf{susp}(A)$ to B, which preserves all the constructors. Expressing this formally, we have the following recursion rule

$$\frac{\Gamma \vdash B : \mathcal{U} \qquad \Gamma \vdash M : \mathsf{susp}(A) \qquad \Gamma \vdash b_N : B \qquad \Gamma \vdash b_S : B \qquad \Gamma, x : A \vdash m(x) : b_N =_B b_S}{\Gamma \vdash \mathsf{rec}[B](b_N; b_S; x.m(x))(M) : B} (\mathsf{susp}(A)\mathsf{rec})$$

Furthermore, the recursor behaves according to the following computation rules:

$$\mathsf{rec}[B](b_N; b_S; x.m(x))(N) \equiv b_N : B$$
$$\mathsf{rec}[B](b_N; b_S; x.m(x))(S) \equiv b_S : B$$
$$\mathsf{ap}_{\mathsf{rec}[B](b_N; b_S; x.m(x))(-)}(\mathsf{mer}(a)) =_{b_S =_B b_N} m(a)$$

The first two computation rules can be considered as β-rules for the 0-cells and the last one a β-rule for the 1-cells. Since we do not care about the specific witness term for the propositional equality in the conclusion of the last rule, we simply omit the witness.

We have an analogous induction schema, where instead of simple types $B : \mathcal{U}$ we consider dependent types $E : \mathsf{susp}(A) \to \mathcal{U}$. The induction rule states that in order to

construct a dependent function mapping $z : \mathsf{susp}(A)$ to an element of $E(z)$, it suffices to give elements $e_N : E(N)$ and $e_S : E(S)$ such that for each $x : A$, e_N and e_S are associated over the path $\mathsf{mer}(x)$. Formally, this means we have the following induction rule:

$$\frac{\begin{array}{c} \Gamma, z : \mathsf{susp}(A) \vdash E(z) : \mathcal{U} \\ \Gamma \vdash M : \mathsf{susp}(A) \qquad \Gamma \vdash e_N : E(N) \qquad \Gamma \vdash e_S : E(S) \\ \Gamma \vdash, x : A \vdash m(x) : e_N =^{z.E(z)}_{\mathsf{mer}(x)} e_S \end{array}}{\Gamma \vdash \mathsf{ind}[z.E(z)](e_N; e_S; x.m(x))(M) : E(M)}$$

Likewise, we have the following computation rules:

$$\mathsf{ind}[z.E(z)](e_N; e_S; x.m(x))(N) \equiv e_N : E(N)$$
$$\mathsf{ind}[z.E(z)](e_N; e_S; x.m(x))(S) \equiv e_S : E(S)$$
$$\mathsf{dap}_{\mathsf{ind}[z.E(z)](e_N; e_S; x.m(x))(-)}(\mathsf{mer}(a)) =_{e_N =^{z.E(z)}_{\mathsf{mer}(x)} e_S} m(a)$$

where the conclusion of the last rule refers to the application of a dependent function to a path, denoted by dap. We can state and prove a useful uniqueness principle, also known as the η-rule, asserting that "if a function behaves like the inductor, then it must be the inductor". We leave the exact statement of this principle and its proof as an exercise.

Why are we interested in suspensions in the first place? Interestingly, many familiar (and also not so familiar) inductive types can be characterized as suspensions. For example:

Exercise. *Show that the type* $\mathsf{susp}(0)$ *is equivalent to the type* **2**.

Exercise. *Show that the type* $\mathsf{susp}(1)$ *is equivalent to the interval type* I.

What is $\mathsf{susp}(2)$? Since the type **2** contains only two elements (up to homotopy), we can picture $\mathsf{susp}(2)$ as a type generated by the two points N and S with two distinct paths between them, called w and e:
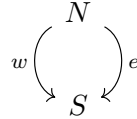
This of course looks very much like a circle - and indeed it is!

Exercise. *Show that the type* $\mathsf{susp}(2)$ *is equivalent to the circle type* $\mathbb{S}^1$.

Now we can ask the question, what is $\mathsf{susp}(\mathsf{susp}(2))$? The type $\mathsf{susp}(\mathsf{susp}(2))$ will of course have to contain the two points N and S. A function from $\mathsf{susp}(2)$ to the path space $N = S$ can then be thought of as a quadruple (w, e, γ, δ), where w, e are two distinct paths from N to S and γ, δ are two distinct paths from w to e. The type $\mathsf{susp}(\mathsf{susp}(2))$ generated by all this data can then be visualized as in Fig. 1:

This looks very much like a sphere - hence we can simply make the definition $\mathbb{S}^2 :\equiv \mathsf{susp}(\mathbb{S}^1)$. We can iterate this and set $\mathbb{S}^{n+1} :\equiv \mathsf{susp}(\mathbb{S}^n)$.
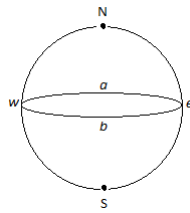
Figure 1: $\mathbb{S}^2$