Overview on Formalization - Type Theory part 3

George McNinch

2024-04-08

Inductive Types

Here I mostly want to "explain a little via some examples", referring you to the text (Rijke n.d.) for more details and full definitions.

The main point to emphasize is that *inductive types* share some of the features of the Π-types – i.e. types of (dependent or ordinary) functions.

Inductive types are specified by: - *constructors* AKA *introduction rules* - an *induction principle* AKA *elimination rules* - *computation rules*

Natural numbers N

- *formation rule*

$$\frac{}{\vdash \mathbb{N} \text{ type}}$$

- *introduction rules* (constructors)

$$\frac{}{\vdash 0_{\mathbb{N}} : \mathbb{N}}$$

$$\frac{}{\vdash \succ : \mathbb{N} \to \mathbb{N}}$$

- induction principle

$$\frac{\begin{array}{c}\Gamma, n : \mathbf{N} \vdash P(n) \text{ type} \\ \Gamma \vdash p_0 : P(0_{\mathbf{N}}) \\ \Gamma \vdash p_S : \Pi_{(n:\mathbf{N})}(P(n) \to P(\succ n))\end{array}}{\Gamma \vdash \text{ind}_{\mathbf{N}}(p_0, p_S) : \Pi_{(n:\mathbf{N})} P(n)} \ \mathbf{N}\text{-ind}$$

List A

Let's describe the inductive type of *Lists*. More precisely, for a type A, let's describe the type whose elements are *lists* of elements of A.

- *formation rule*

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \text{List } A \text{ type}} \ \text{List-formation}$$

- *introduction rules*; ie. the *constructors*.

  There are two introduction rules for lists. The first one forms the *empty list*:

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \text{nil} : \text{List } A}$$

  The second one constructs ("`cons`") a list from a term of type A and an existing List A.

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \text{cons} : A \to \text{List}\, A \to \text{List}\, A}$$

- the induction principle or elimination rule

  We must stipulate what is needed to construct a section of a type family over the inductive type – i.e. a dependent function.

  The idea is this: for our inductive type A, in order to define a dependent function $f : \Pi_{(x:A)} B(x)$ one must specify the behavior of f at the constructors of A.

  Here is our induction principle for List A:

$$\frac{\begin{array}{c}\Gamma, l : \text{List}(L) \vdash P(l) \text{ type} \\ \Gamma \vdash p_{\text{nil}} : P(\text{nil}) \\ \Gamma \vdash p\,\text{cons} : \Pi_{(l:\text{List}\, A)}(A \to P(x) \to P(\succ n))\end{array}}{\Gamma \vdash \text{ind}\mathbf{N}(p_0, p_S) : \Pi_{(n:\mathbf{N})} P(n)} \; \mathbf{N}\text{-ind}$$

(iii) The computation rules assert that the inductively defined section agrees on the constructors with the data that was used to define the section. Thus, there is a computation rule for every constructor

Bibliography

Bibliography

Rijke, Egbert. n.d. “Introduction to Homotopy Type Theory.” https://arxiv.org/abs/2212.11082. Accessed April 10, 2024.