Draft

Dimitrios Koutsoulis
11838639

May 7, 2019

1 Type Theory

1.1 Introduction

Type theory is a formal language and deductive system, that is self sufficient in the sense that it need not be formulated as a collection of axioms on top of some other formal system like First Order Logic, instead its deductive system can be built on top of its own formal language.

Central to Type Theory is the notion of *Type*. Every construction a in Type Theory we come across, must lie in some type A, which we denote as $a : A$. Note that the relation $:$ is transitive, so $a : A$ and $A : B$ imply $a : B$. This induces the cummulative hierarchy shape of the universe of types $\mathcal{U}$, wherein every level of the universe includes all lower levels and their types.

For the deductive part of Type Theory, we interpret propositions as types. Proving proposition P amounts to providing some inhabitant $p : P$.

1.2 Type Construction Operations

Let's have a look at some important type constructions.

- Given types $A, B : \mathcal{U}$ we can define the type $A \to B$ of functions from A to B. We can use λ-abstraction to construct elements of this type. $\lambda x.\Phi$ lies in $A \to B$ iff for $a : A$ we have $\Phi[a/x] : B$. For $f : A \to B$ and $a : A$ we have that the application of f on a, denoted as $f(a)$ or $f\ a$, lies in B.
- Given some type $A : \mathcal{U}$ and a family of types B over A, $B : A \to \mathcal{U}$, we have the type of dependent functions

$$\prod_{a:A} B(a)$$

where for $f : \prod_{a:A} B(a)$ and $x : A$ we have $f(x) : B(x)$. As in the case of non-dependent functions, we can use lambda abstraction to construct elements of a dependently-typed function type.

- Given $A, B : \mathcal{U}$ we can define the product type $A \times B : \mathcal{U}$. For $a : A$ and $b : B$ we have the pair $(a, b) : A \times B$. We also have the projection functions

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

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

- Given $A : \mathcal{U}$ and family of types B over A, $B : A \to \mathcal{U}$, we can define the dependent pair type

$$\sum_{a:A} B(a)$$

Given $x : A$ and $b : B(x)$ we can construct the pair $(x, b) : \sum_{a:A} B(a)$. We have two projection functions, similar to the case of the product type.

- Given $A, B : \mathcal{U}$ we can construct the coproduct type $A + B$. We can construct elements of $A + B$ using the functions

$$\mathrm{inl} : A \to A + B$$

$$\mathrm{inr} : B \to A + B$$

This induces the induction principle

$$\mathrm{ind}_{A+B} : \prod_{C:A+B\to U} \Big(\prod_{a:A} C(\mathrm{inl}\ a) \Big) \to \Big(\prod_{b:A} C(\mathrm{inr}\ b) \Big) \to \prod_{x:A+B} C(x)$$

- Given $x, y : A$ we have the **identity type** $x =_A y$. An element of this type amounts to a proof that x and y are equal. Say x and y are judgmentally equal. This is captured by the element $\mathrm{idp}_x : x =_A y$. The relevant induction principle describes how we can use elements of an identity type

$$\mathrm{ind}_{=_A} : \prod_{C:\prod_{x,y:A}(x=_A y)\to\mathcal{U}} \Big(\prod_{x:A} C(x, x, \mathrm{idp}_x) \Big) \to \prod_{x,y:A} \prod_{(p:x=_A y)} C(x, y, p)$$

We can concatenate those paths whose domains and codomains allow for it. Paths are equivalences. That is if $p : x = y$ is such a path, we can provide its inverse p^{-1} for which we have in turn a path between $p \cdot p^{-1}$ and idp_y and another one between $p^{-1} \cdot p$ and idp_x.

We are now well-equipped to view types as $(\infty, 1)$-categories.
Under this view

- elements of the type are objects (or 0-morphisms) of the category
- identity paths between elements are 1-morphisms
- paths between n-paths are $(n + 1)$-morphisms
- all paths are reversible

2 LLPO

Definition 2.1. *The Lesser Limited Principle of Omniscience, states that given binary sequence* $s : \mathbb{N} \to \mathbf{2}$ *and the fact that there is at most one occurence of* 1 *in the sequence, formally*

$$\text{atMost1one} : \prod_{n_1:\mathbb{N}} \prod_{n_2:\mathbb{N}} s(n_1) = 1 \to s(n_2) = 1 \to n_1 = n_2$$

we can then have by the LLPO a witness for $p_{\text{odd}} \vee p_{\text{even}}$*, where* p_{odd} *is the statement that for all odd positions* n*,* $s(n) = 0$*, formally* $p_{\text{odd}} \equiv \prod_{n:\mathbb{N}} \text{odd}(n) \to s(n) = 0$*. Similarly for* p_{even}*.*

LLPO can be viewed as a weaker form of the Law of Excluded Middle.
Lemma 2.2. *The Law of Excluded Middle implies the Limited Principle of Omniscience.*

Proof. By LEM we have a witness $l_1 : p_{\text{odd}} \vee \neg p_{\text{odd}}$. Since this is a coproduct, by the relevant principle of induction, it's enough to prove LLPO from the disjuncts.

- $p_{\text{odd}} \Rightarrow$ LLPO, trivially.
- $\neg p$, alongside LEM, implies that there exists odd $n_e : \mathbb{N}$ such that $s(n_e) = 1$. We can now provide $\prod_{n:\mathbb{N}} even(n) \to s(n) = 0$. Let $n : \mathbb{N}$. By the definition of s, $s(n) = 0 \vee s(n) = 1$. We invoke the principle of induction of coproducts and prove LLPO from the disjuncts.
  - $s(n) = 0$, in this case we are done.
  - $s(n) = 1$. By atMost1one we have $n = n_e \Rightarrow n$ even and odd which is a contradiction $c : \bot$. By LEM we have Reductio Ad Absurdum raa : $\bot \to s(n) = 0$. Then raa$(c) : s(n) = 0$.

□

3 Modalities

Definition 3.1. *A* ***modality*** *is a function* $\bigcirc : \mathcal{U} \to \mathcal{U}$ *with the following properties.*

1. *For every type* A *we have a function* $\eta^{\bigcirc}_A : A \to \bigcirc A$
2. *for every* $A : \mathcal{U}$ *and every type family* $B : \bigcirc A \to \mathcal{U}$ *we have a function*

$$ind_{\bigcirc} : \Big(\prod_{a:A} \bigcirc(B(\eta^{\bigcirc}_A\, a)) \Big) \to \prod_{z:\bigcirc A} \bigcirc(B\ z)$$

3. *For every* $f : \prod_{a:A} \bigcirc(B(\eta^{\bigcirc}_A\, a))$ *there is a path* $ind_{\bigcirc}(f)(\eta^{\bigcirc}_A\, a) = f\ a$
4. *For all* $z, z' : \bigcirc A$*, the function* $\eta^{\bigcirc}_{z=z'} : (z = z') \to \bigcirc(z = z')$ *is an equivalence.*

A modality $\bigcirc$ induces a Σ-closed reflective subuniverse.

Definition 3.2. *Given modality* $\bigcirc : \mathcal{U} \to \mathcal{U}$*, a* ***reflective subuniverse*** *is a "subset" of* $\mathcal{U}$ *encoded by a family of h-propositions* $P : \mathcal{U} \to \mathrm{Prop}$ *such that the following conditions hold.*

- *For* $A : \mathcal{U}$*, we have* $P(\bigcirc A)$*.*
- *For* $A : \mathcal{U}$ *and* B *such that* $P(B)$*, the function*

$$\lambda f.f \circ \eta_A : (\bigcirc A \to B) \to (A \to B)$$

  is an equivalence.

The subuniverse is Σ***-closed*** *if for* X *such that* $P(X)$ *and* $Q : X \to \mathcal{U}$ *such that* $\prod_{x:X} P(Q(x))$*, we have* $P(\Sigma_{x:X} Q(x))$*.*